

ASTRONOMICAL DATA ANALYSIS SOFTWARE AND SYSTEMS XXVIII

COVER ILLUSTRATION:

Screen capture from the winning hackathon entry, *Music of Light Curves*, a sonification of variable stars from the Gaia catalogue. See also Thomas et al., p.723

Credit: Thomas Boch, Matthieu Baumann, and Siddha Mavuram

ASTRONOMICAL SOCIETY OF THE PACIFIC
CONFERENCE SERIES

A SERIES OF BOOKS ON RECENT DEVELOPMENTS IN ASTRONOMY AND ASTROPHYSICS

Volume 523

EDITORIAL STAFF

Managing Editor: Joseph Jensen
Associate Managing Editor: Jonathan Barnes
Publication Manager: Cindy Moody
Editorial Assistant: Blaine Haws
Publication Consultant: Pepita Ridgeway
e-Book Specialist: Cicely Potter

MS 179, Utah Valley University, 800 W. University Parkway, Orem, Utah 84058-5999

Phone: 801-863-8804 E-mail: aspcs@aspbooks.org

E-book site: <http://www.aspbooks.org>

PUBLICATION COMMITTEE

Jeff Mangum, co-Chair
National Radio Astronomy Observatory

Joseph Jensen, co-Chair
Utah Valley University

Bruce Elmegreen
IBM Watson Research Center

Lynne Hillenbrand
California Institute of Technology

Doug Leonard
San Diego State University

Chris Packham
University of Texas at San Antonio

Amy Mainzer
Jet Propulsion Laboratory

ASPCS volumes may be found online with color images at <http://www.aspbooks.org>.

ASP Monographs may be found online at <http://www.aspmonographs.org>.

For a complete list of ASPCS Volumes, ASP Monographs, and
other ASP publications see <http://www.astro society.org/pubs.html>.

All book order and subscription inquiries should be directed to the ASP at
800-335-2626 (toll-free within the USA) or 415-337-2126,
or email service@astro society.org

ASTRONOMICAL SOCIETY OF THE PACIFIC
CONFERENCE SERIES

Volume 523

**ASTRONOMICAL DATA ANALYSIS SOFTWARE AND
SYSTEMS XXVIII**

Proceedings of a conference held at
The Hotel at the University of Maryland, College Park, Maryland, USA
11–15 November 2018

Edited by

Peter J. Teuben

Astronomy Department, University of Maryland, College Park, MD, USA

Marc W. Pound

Astronomy Department, University of Maryland, College Park, MD, USA

Brian A. Thomas

Office of Chief Information Officer, NASA HQ, Washington, DC, USA

Elizabeth M. Warner

Astronomy Department, University of Maryland, College Park, MD, USA



SAN FRANCISCO

ASTRONOMICAL SOCIETY OF THE PACIFIC

390 Ashton Avenue
San Francisco, California, 94112-1722, USA

Phone: 415-337-1100

Fax: 415-337-5205

E-mail: service@astrosociety.org

Web site: www.astrosociety.org

E-books: www.aspbooks.org

First Edition

© 2019 by Astronomical Society of the Pacific

ASP Conference Series

All rights reserved.

No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means—graphic, electronic, or mechanical, including photocopying, taping, recording, or by any information storage and retrieval system—without written permission from the Astronomical Society of the Pacific.

ISBN: 978-1-58381-933-3

e-book ISBN: 978-1-58381-934-0

Library of Congress (LOC) Cataloging in Publication (CIP) Data:

Main entry under title

Library of Congress Control Number (LCCN): ISSN: 1080-7926; LCCN: 2019905912

Printed in the United States of America by Sheridan Books, Ann Arbor, Michigan.

This book is printed on acid-free paper.

Dedication

It is with great sadness that I report the death of Jim Lewis who was an avid attendee of the ADASS conferences over many years and a keen member of the POC for the last 14 years. Jim loved taking a full part in ADASS gatherings and was heartbroken when due to ill health he had to miss one meeting and lose his coveted “Friends of Betty” gold star.

I first met Jim when he came to the Institute of Astronomy in 1986, after completing his PhD at Mt. Stromlo under Ken Freeman, and worked closely with him over the majority of the intervening years. Jim was involved in many projects including data reduction pipelines for the WYFFOS spectrograph on the WHT; the Wide Field Camera on the INT; which lead to the development of the VISTA Data Flow System for processed images taken with WFCAM on UKIRT and VIRCAM on VISTA; and more recently on the deployment and development of the Gaia-ESO, WEAVE and 4MOST spectroscopic data analysis systems.

Jim was a wonderful human being and a great colleague. We will remember him not only for his incredible depth of knowledge, but also for his wonderful personality, his keen sense of humour and his ability to quote whole passages of Fawlty Towers without pause. We are grateful to have had him as our colleague and friend and we will miss him dearly as we know many whose lives he touched will.

Jim had been suffering from cancer for many years, but in spite of his periods of ill health he was always determined to continue working on all the projects he was involved in and did so up to a few days before he died on Easter Sunday, 21st April 2019. Jim was buried in the churchyard at St. Marys in Hardwick resting peacefully in the middle of the community he was a major part of. He was 59 and is survived by Carole and their two children Jess and Jacob.

Mike Irwin
Director
Cambridge Astronomy Survey Unit



Jim Lewis at ADASS XXVII in Santiago, Chile.

Contents

Dedication	v
<i>M. Irwin</i>	
Preface	xxi
<i>P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner</i>	
Participants	xxv
Conference Photograph	xxxiii

Session I. Astrophysical Data Visualization from Line Plots to Augmented and Virtual Reality

3D Data Visualization in Astrophysics	3
<i>B. R. Kent (Invited Speaker)</i>	
An Introduction to FITSWebQL	13
<i>C. Zapart, Y. Shirasaki, M. Ohishi, Y. Mizumoto, W. Kawasaki, T. Kobayashi, G. Kosugi, E. Morita, A. Yoshino, and S. Eguchi</i>	
An HDF5 Schema for SKA Scale Image Cube Visualization	17
<i>A. Comrie, A. Pińska, R. Simmonds, and A. R. Taylor</i>	
Analysis of Astronomical Data using VR: the Gaia Catalog in 3D	21
<i>E. Ramirez, J. González-Núñez, J. Hernandez, J. Salgado, A. Mora, U. Lammers, B. Merin, D. Baines, G. de Marchi, and C. Arviset</i>	
Exoplanets Data Visualization in Multi-dimensional Plots using Virtual Reality in DACE	25
<i>F. Alesina, F. Cabot, N. Buchschacher, and J. Burnier</i>	
VisIVO Visual Analytics Tool: An EOSC Science Demonstrator for Data Discovery	29
<i>U. Becciani, F. Vitello, E. Sciacca, A. Costa, A. Calanducci, S. Riggi, and S. Molinari</i>	
Open-source Web Tools for Spectroscopic and Imaging Data Visualization for the VOXAstro Initiative	33
<i>K. A. Grishin, I. Chilingarian, and I. Katkov</i>	
Vissage: Viewing Polarization Data from ALMA	37
<i>W. Kawasaki, Y. Shirasaki, C. Zapart, A. Yoshino, E. Morita, T. Kobayashi, G. Kosugi, M. Ohishi, and Y. Mizumoto</i>	
Realtime Telescope Data Visualization using Web Technologies	41
<i>P. Mellado</i>	

viii	<i>Contents</i>	
TOPCAT and Gaia		43
	<i>M. B. Taylor</i>	
 Session II. Machine Learning in Astronomy		
Deep Learning of Astronomical Features with Big Data		49
	<i>M. Lieu, D. Baines, G. Fabrizio, B. Merin, C. Arviset, B. Altieri, L. Conversi, and B. Carry (Invited Speaker)</i>	
Automatic Classification of Transiting Planet Candidates using Deep Learning . .		59
	<i>M. Ansdell, Y. Ioannou, H. P. Osborn, M. Sasdelli, J. C. Smith, D. Caldwell, J. M. Jenkins, C. Räissi, and D. Angerhausen</i>	
Acceleration of Non-Linear Minimization with PyTorch		63
	<i>B. Nikolic</i>	
Feature Selection for Better Spectral Characterization or: How I Learned to Start Worrying and Love Ensembles		67
	<i>S. Gilda</i>	
A Method to Detect Radio Frequency Interference Based on Convolutional Neural Networks		71
	<i>C. Dai, S. F. Zuo, W. Liu, J. X. Li, M. Zhu, F. Q. Wu, and X. C. Yu</i>	
Cherenkov Shower Detection Combining Probability Distributions from Convolutional Neural Networks		75
	<i>M. Araya, F. Casas, and R. Cáseres</i>	
A New Implementation of Deep Neural Network for Spatio-Spectral Analysis in X-Ray Astronomy		79
	<i>H. Iwasaki, Y. Ichinohe, Y. Uchiyama, and H. Yamaguchi</i>	
Variable Star Classification using Multi-View Metric Learning		83
	<i>K. B. Johnston, S. M. Caballero-Nieves, A. M. Peter, V. Petit, and R. Haber</i>	
Multiscale Spatial Analysis of Young Stars Complex using the dbscan Clustering Algorithm		87
	<i>I. Joncour, E. Moraux, G. Duchêne, and L. G. Mundy</i>	
LAMOST DR5 Spectral Clustering for Stellar Templates Construction		91
	<i>X. Kong and A.-L. Luo</i>	
Saving Endangered Animals with Astro-Ecology		95
	<i>P. McWhirter, J. Veitch-Michaelis, C. Burke, M. Lam, and S. N. Longmore</i>	
MaxiMask: Identifying Contaminants in Astronomical Images using Convolutional Neural Networks		99
	<i>M. Paillassa, E. Bertin, and H. Bouy</i>	
A Hybrid Neural Network Approach to Estimate Galaxy Redshifts from Multi-Band Photometric Surveys		103
	<i>R. D. C. dos Santos, F. C. de Souza, A. Muralikrishna, and W. A. dos Santos Junior</i>	

Contents	ix
Chatting with the Astronomical Data Services	107
<i>A. Schaaff, A. Guyot, T. Boch, and S. Derriere</i>	
Machine Learning from Cosmological Simulations to Identify Distant Galaxy Mergers	111
<i>G. Snyder</i>	
A Machine Learning Approach for Dark-Matter Particle Identification Under Extreme Class Imbalance	115
<i>R. Sutrisno, R Vilalta, and A Renshaw</i>	
Analysis of Stellar Spectra from LAMOST DR5 with Generative Spectrum Net- works	119
<i>R. Wang and A.-L. Luo</i>	
U-NetIM: An Improved U-Net for Automatic Recognition of RFIs	123
<i>M. Long, Z.. Yang, J. Xiao, C. Yu, and B. Zhang</i>	
Automatic Detection of Microlensing Events in the Galactic Bulge using Machine Learning Techniques	127
<i>S. Chu, K. L. Wagstaff, G. Bryden, and Y. Shvartzvald</i>	
 Session III. Data Science: Workflows, Hardware, Software, Humanware	
Massive Data Exploration in Astronomy: What Does Cognitive Have To Do With It?	133
<i>K. Borne (Invited Speaker)</i>	
A New Synthesis Imaging Tool for ALMA Based on Sparse Modeling	143
<i>T. Nakazato, S. Ikeda, K. Akiyama, G. Kosugi, M. Yamaguchi, and M. Honma</i>	
Towards New Solutions for Scientific Computing: The Case of Julia	147
<i>M. Tomasi and M. Giordano</i>	
Performance Analysis of the SO/PHI Software Framework for On-board Data Reduction	151
<i>K. Albert, J. Hirzberger, D. Busse, J. B. Rodríguez, J. S. C. Durán, J. P. C. Carrascosa, B. Fiethe, A. Gandorfer, Y. Guan, M. Kolleck, A. Lagg, T. Lange, H. Michalik, S. K. Solanki, J. C. d. T. Iniesta, and J. Woch</i>	
Optimization of Aperture Photometry in the Chandra Source Catalog	155
<i>C. Allen, J. Miller, and F. Primini</i>	
Utilizing Conda for Fermi Data Analysis Software Releases	159
<i>J. Asercion</i>	
Streamlining Pipeline Workflows: Using Python with an Object-Oriented Approach to Consolidate Aggregate Pipeline Processes	163
<i>M. Brown, J. A. Mader, G. B. Berriman, C. R. Gelino, M. Kong, A. C. Laity, J. Riley, L. Rizzi, and M. A. Swain</i>	
GMRT Archive Processing Project	167
<i>S. Deshpande, Y. Wadadekar, H. Intema, B. Ratnakumar, L. George, R. Desai, A. Sakhadeo, S. Shaikh, C. H. Ishwara-Chandra, and D. Oberoi</i>	

x	<i>Contents</i>
Computational Astrophysics with Go	171
<i>P. Gupta</i>	
JWST Data Management Subsystem Operations: Rehearsing to Receive, Process, and Archive JWST Data	175
<i>C. Kaleida, A. Alexov, M. Kyprianou, F. Abney, and M. Burger</i>	
CIAO: A Look Under the Hood of Chandra’s X-Ray Imaging and Analysis Software Configuration Management – Past, Present, and Future.	179
<i>Z. Kaufman, M. Cresitello-Dittmar, J. D. Evans, O. Laurino, W. McLaughlin, and J. Miller</i>	
DALiUGe/CASA Based Processing for the Extragalactic HI Observations with FAST	183
<i>V. Kitaeff, M. Zhu, L. Staveley-Smith, R. Tobar, K. Vinsen, A. Wicenec, and C. Wu</i>	
Running GTC Data Reduction Pipelines in Jupyter	187
<i>S. Pascual, N. Cardiel, C. Cabello, M. Camorro-Cazorla, C. Catalán-Torrecilla, B. T. Dullo, Á. Castillo-Morales, A. Gil de Paz, and J. Gallego</i>	
Reprocessing All the XMM-Newton Scientific Data: A Challenge for the Pipeline Processing System	191
<i>J. Perea-Calderon, P. Rodriguez-Pascual, and C. Gabriel</i>	
MeerKAT: Operational Workflow and Data Analysis	195
<i>R. Renil</i>	
Euclidizing External Tools: An Example from SDC-IT on How to Handle Software and Humanware	199
<i>E. Romelli, M. Frailis, S. Galeotta, D. Tavagnacco, D. Maino, C. Vuerli, G. Maggio, and G. Taffoni</i>	
A Real-Time Data Reduction Pipeline for the Goodman Spectrograph	203
<i>S. Torres and C. Briceño</i>	

Session IV. Management of Large Science Projects

Hit the Ground Running: Data Management for JWST	209
<i>A. Alexov, M. Kyprianou, and C. Kaleida</i>	
Gaia DPAC Project Office: Coordinating the Production of the Largest Star Catalogue	213
<i>G. Gracia-Abril, D. Teyssier, J. Portell, A. Brown, A. Vallenari, F. Jansen, and U. Lammers</i>	
The VLA Sky Survey – Operations, Data Processing and Archiving	217
<i>M. Lacy, C. Chandler, A. Kimball, S. Myers, K. Nyland, and S. Witz</i>	

Session V. Science Platforms: Tools for Data Discovery and Analysis from Different Angles

Hubble in the Cloud: A Prototype of a Science Platform at STScI	223
<i>I. Momcheva and A. Smith (Invited Speaker)</i>	
The NOAO Data Lab: Design, Capabilities, and Community Development	233
<i>M. Fitzpatrick, K. Olsen, G. Eychaner, L. Fulmer, L. Huang, S. Juneau, D. Nidever, R. Nikutta, and A. Scott</i>	
Astropy and the Virtual Observatory	237
<i>T. Donaldson</i>	
Lilith: A Versatile Instrument and All-Sky Simulator and its Application to TESS	241
<i>J. C. Smith, P. Tenenbaum, J. M. Jenkins, and J. D. Twicken</i>	
AFLAK: Visual Programming Environment with Quick Feedback Loop, Tuned for Multi-Spectral Astrophysical Observations	245
<i>M. O. Boussejra, S. Takekawa, R. Uchiki, K. Matsubayashi, Y. Takeshima, M. Uemura, and I. Fujishiro</i>	
DEAVI: Dynamic Evolution Added Value Interface	249
<i>D. Baines, I. de la Calle, J. M. Herrera-Fernandez, A. Ibarra, J. Salgado, and L. Valero-Martin</i>	
New Python Developments to Access CDS Services	253
<i>M. Baumann and T. Boch</i>	
Breathing New Life into an Old Pipeline: Precision Radial Velocity Spectra of TESS Exoplanet Candidates	257
<i>G. B. Berriman, D. Ciardi, B. J. Fulton, J. C. Good, M. Kong, H. Isaacson, and J. Walawender</i>	
Gaia Photometric Science Alerts Data Flow	261
<i>A. Delgado, S. Hodgkin, D. W. Evans, D. L. Harrison, G. Rixon, F. van Leeuwen, M. van Leeuwen, and A. Yoldas</i>	
The CASA Software for Radio Astronomy: Status Update from ADASS 2018 . .	265
<i>B. Emonts, R. Raba, F. M. Pouzols, T. Tsutsumi, T. Nakazato, A. Kepley, D. Schiebel, S. Castro, A. Comrie, K.-S. Wang, S. Bhatnagar, P. Brandt, C. Brogan, J. D. Meyer, P. Ford, K. Golap, C. E. García-Dabó, B. Garwood, A. Hale, T. Hunter, B. R. Kent, W. Kawasaki, R. Indebetouw, D. Mehringer, R. Miel, G. Moellenbrock, S. Nishie, J. Ott, D. Petry, M. Pokorny, U. Rau, C. Reynolds, K. Sugimoto, V. Suoranta, N. Schweighart, D. Tafoya, A. Wells, and I. Yoon</i>	
Data Analysis Tools for JWST and Beyond	269
<i>H. Ferguson</i>	
Optimization Strategies for Running Legacy Codes	273
<i>J. Lammers and P. J. Teuben</i>	
Arcade: An Interactive Science Platform in CANFAR	277
<i>B. Major, J. Kavelaars, S. Fabbro, D. Durand, and H. Jeeves</i>	

xii	<i>Contents</i>	
Exploring Space, Time, and Data with WCSTools		281
	<i>J. Mink</i>	
ESAC Science Exploitation and Preservation Platform Reference Architecture . .		285
	<i>V. Navarro, R. Alvarez, F. Pérez-López, C. Arviset, J. Ventura-Traveset, and A. Martín Furones</i>	
Stellar Atmospheric Parameters from Full Spectrum Fitting of Intermediate- and High-resolution Spectra against PHOENIX/BT-Settl Synthetic Stellar Atmospheres		289
	<i>E. Rubtsov, I. Chilingarian, S. Borisov, and I. Katkov</i>	
 Session VI. Quality Assurance of Science Data		
The BagIt Packaging Standard for Interoperability and Preservation		295
	<i>R. Plante, G. Greene, and R. Hanisch (Invited Speaker)</i>	
Adding Science Validation to the JWST Calibration Pipeline		305
	<i>R. I. Díaz and M. M. García Marín</i>	
Quality Assurance in the Ingestion of Data into the CDS Vizier Catalogue and Data Services		309
	<i>G. Landais, P. Ocvirk, M. G. Allen, M. Brouty, E. Perret, T. Pouvreau, and P. Vannier</i>	
ProvTAP: A TAP Service for Providing IVOA Provenance Metadata		313
	<i>F. Bonnarel, M. Louys, G. Mantelet, M. Nullmeier, M. Servillat, K. Riebe, and M. Sanguillon</i>	
Rectification and Wavelength Calibration of EMIR Spectroscopic Data with Python		317
	<i>N. Cardiel, S. Pascual, J. Gallego, C. Cabello, F. Garzón, M. Balcells, N. Castro-Rodríguez, L. Domínguez-Palmero, P. Hammersley, L. R. Patrick, R. Pelló, M. Prieto, and A. Streblyanska</i>	
DRAGONS – Data Reduction for Astronomy from Gemini Observatory North and South		321
	<i>K. Labrie, K. Anderson, R. Cárdenes, C. Simpson, and J. E. H. Turner</i>	
stginga: Ginga Plugins for Data Analysis and Quality Assurance of HST and JWST Science Data		325
	<i>P. L. Lim and E. R. Jeschke</i>	
A Triplestore Implementation of the IVOA Provenance Data Model		329
	<i>M. Louys, F.-X. Pineau, F. Bonnarel, and L. Holzmann</i>	
The IVOA Provenance Data Model		333
	<i>M. Servillat, K. Riebe, F. Bonnarel, A. Galkin, M. Louys, M. Nullmeier, M. Sanguillon, and O. Streicher</i>	

Session VII. DevOps Practices in Astronomy Software

DevOps: A Perfect Ally for Science Operations for Large and Distributed Astronomy Projects like Gaia

339

R. Guerra, N. Cheek, E. Anglada, P. Esquej, E. Fraile, E. Pozo, and U. Lammers

(Invited Speaker)

Agile and DevOps from the Trenches at ASTRON

349

G. Loose

Fundamentals of Effective Cloud Management for the New NASA Astrophysics Data System

353

S. Blanco-Cuaresma, A. Accomazzi, M. J. Kurtz, E. Henneken, C. S. Grant, D. M. Thompson, R. Chyla, S. McDonald, G. Shapurian, T. W. Hostetler, M. R. Templeton, K. E. Lockhart, K. Bukovi, and N. Rapport

Versioned Executable User Documentation for In-development Science Tools

357

C. Boisson, J. E. Ruiz, C. Deil, A. Donath, and B. Khelifi

Development, Tests, and Deployment of Web Application in DACE

361

J. Burnier, F. Alesina, and N. Buchschacher

GDL – GNU Data Language 0.9.9

365

A. Coulais, G. Duvert, G. Jung, S. Arabas, S. Flinois, and A Si Lounis

Automating Multimission Access: Rolling Out a Flexible Virtual Observatory-based Infrastructure

369

T. Dower and B. Shiao

Application of Google Cloud Platform in Astrophysics

373

M. Landoni, G. Taffoni, A. Bignamini, and R. Smareglia

Transforming Science Code into Maintainable Software, Insights into the G-CLEF Exposure Time Calculator (ETC)

377

C. Paxson, J. Miller, and J. D. Evans

Centralization and Management of Science Operations Procedures and Test Cases using SOCCI

381

F. Pérez-López, V. Navarro, K. Lumi, H. Liiva, K. Hanson, R. Caballero, C. L. Kuik, A. Lember, C. Garcia, E. Pasenkov, and L. Kaldamae

Session VIII. Databases and Archives: Challenges and Solutions in the Big Data Era

Astronomical Archives: Serving Up the Universe

387

F. Stoehr (Invited Speaker)

Astrocut: A Cutout Service for TESS Full-Frame Image Sets

397

C. Brasseur, C. Phillip, J. Hargis, S. Mullally, S. Fleming, M. Fox, and A. Smith

AXS: Making End-User Petascale Analyses Possible, Scalable, and Usable

401

P. Zečević, C. T. Slater, M. Juric, and S. Lončarić

xiv	<i>Contents</i>	
No-SQL Databases: An Efficient Way to Store and Query Heterogeneous Astronomical Data in DACE		405
<i>N. Buchschacher, F. Alesina, and J. Burnier</i>		
Data-driven Space Science at ESAC Science Data Centre		409
<i>B. Martinez, I. Barbarisi, J. González-Núñez, M. Fernandez, C. Laantee, B. Merin, S. Nieto, H. Perez, J. Salgado, and P. de Teodoro</i>		
Bringing Together the Australian Sky - Coordination and Interoperability Challenges of the All-Sky Virtual Observatory		413
<i>S. O'Toole and K. Sealey</i>		
Driving Gaia Science from the ESA Archive: DR2 to DR3		417
<i>J. González-Núñez, J. Salgado, R. Gutiérrez-Sánchez, J. C. Segovia, J. Duran, E. Racero, J. Osinde, P. de Teodoro, A. Mora, J. Bakker, U. Lammers, B. Merin, C. Arviset, and F. Aguado-Agelet</i>		
Creating and Managing Very Large HiPS: The Pan-STARRS Case		421
<i>T. Boch and P. Fernique</i>		
Archive-2.0: Metadata and Data Synchronization Between MAST, CADC, and ESAC		425
<i>P. Dowler, M. Arevalo, A. Damian, J. Duran, D. Durand, S. Gaudet, J. Hargis, B. Major, B. McLean, O. Oberdorf, and D. R. Rodriguez</i>		
Mapping Data Models to VOTable		429
<i>O. Laurino, G. Lemson, M. Cresitello-Dittmar, T. Donaldson, and L. Michel</i>		
The New Science Portal and the Programmatic Interfaces of the ESO Science Archive		433
<i>A. Micol, M. Arnaboldi, N. Delmotte, V. Forchí, N. Fourniol, O. Hainaut, U. Lange, M. Kahn Ahmed, L. Mascetti, J. Retzlaff, M. Romaniello, D. Sisodia, C. Spiniello, M. Stellert, F. Stoehr, I. Vera, and S. Zampieri</i>		
Science Exploitation in a Big Data Archive: the Euclid Scientific Archive System		437
<i>S. Nieto, P. de Teodoro, F. Giordano, E. Racero, M. Fernandez, D. Noiret, J. Salgado, B. Altieri, B. Merin, and C. Arviset</i>		
The VLITE Database Pipeline		441
<i>E. Polisensky, E. E. Richards, T. Clarke, W. Peters, and N. E. Kassim</i>		
Gaia DR2 and the Virtual Observatory: VO in Operations New Era		445
<i>J. Salgado, J. González-Núñez, R. Gutiérrez-Sánchez, J. C. Segovia, A. Mora, J. Bakker, T. Boch, M. G. Allen, N. C. Hambly, S. Voutsinas, M. Demleitner, J. Duran, P. de Teodoro, D. Baines, B. Merin, and C. Arviset</i>		
The OV-GSO Data Center		449
<i>M. Sanguillon, J.-M. Glorian, and C. Vastel</i>		
The TESS Science Data Archive		453
<i>D. Swade, S. Fleming, J. M. Jenkins, D. W. Latham, E. Morgan, S. Mullally, and R. Vanderspek</i>		

Session IX. Software for Solar System Astronomy

ESASky: A New Window for Solar System Data Exploration 459
E. Racero, F. Giordano, B. Carry, J. Berthier, J. González-Núñez, H. Norman, D. Baines, B. Merin, B. L. Marti, M. Lopez-Caniego, P. de Teodoro, J. Salgado, and C. Arviset

The PDS Approach to Science Data Quality Assurance 463
A. Rough

Modeling Effects of Stellar UV-Driven Photochemistry on the Transit Spectra of Moist Rocky Atmospheres Around M Dwarfs 467
M. Afrin Badhan, E. T. Wolf, R. K. Kopparapu, G. Arney, E. M.-R. Kempton, D. Deming, and S. Domagal-Goldman

ZChecker: Finding Cometary Outbursts with the Zwicky Transient Facility . . . 471
M. S. P. Kelley, D. Bodewits, Q. Ye, R. R. Laher, F. J. Masci, S. Monkewitz, R. Riddle, B. Rusholme, D. L. Shupe, and M. T. Soumagnac

Session X. Time Domain Astronomy

Data Challenges of the VO in Time Domain Astronomy 477
A. Nebot (Invited Speaker)

The ZTF Alert Stream: Lessons from the First Six Months of Operating an LSST Precursor 485
M. Juric, E. C. Bellm, M. T. Patterson, V. Z. Golkhou, and B. Rusholme

A GPU Implementation of the Harmonic Sum Algorithm 489
K. Adámek and W. Armour

Prototype Implementation of a Web-Based Gravitational Wave Signal Analyzer: SNEGRAF 493
S. Eguchi, S. Shibagaki, K. Hayama, and K. Kotake

Time in Aladin 497
P. Fernique, D. Durand, and A. Nebot

Session XI. Multi-Messenger Astronomy

Coordinating Observations Among Ground and Space-Based Telescopes in the Multi-Messenger Era 503
E. Kuulkers, M. Ehle, C. Gabriel, A. Ibarra, P. Kretschmar, B. Merin, J.-U. Ness, E. Salazar, J. Salgado, C. Sánchez-Fernández, R. Saxton, and E. M. Levesque (Invited Speaker)

Searching for Optical Counterparts to Gravitational Wave Events with the Catalina Sky Survey 511
M. J. Lundquist, D. Sand, E. Christensen, W. Fong, and K. Paterson

xvi	<i>Contents</i>	
	CALET Gamma-ray Burst Monitor Web-analysis System	515
	<i>K. Ebisawa, S. Nakahira, T. Sakamoto, and A. Yoshida</i>	
	Session XII. Algorithms	
	An Overview of the LSST Image Processing Pipelines	521
	<i>J. Bosch, Y. AlSayyad, R. Armstrong, E. C. Bellm, H. Chiang, S. Eggl,</i> <i>K. Findeisen, M. Fisher-Levine, L. P. Guy, A. Guyonnet, Ž. Ivezić, T. Jenness,</i> <i>G. Kovács, K. S. Krughoff, R. H. Lupton, N. B. Lust, J. Meyers, L. MacArthur,</i> <i>F. Moolekamp, C. B. Morrison, T. D. Morton, W. O’Mullane, J. Parejko,</i> <i>A. A. Plazas, P. A. Price, M. L. Rawls, S. Reed, P. Schellart, C. T. Slater,</i> <i>I. Sullivan, J. D. Swinbank, D. Taranu, C. Z. Waters, and W. M. Wood-Vasey</i> <i>(Invited Speaker)</i>	
	Performance-related Aspects in the Big Data Astronomy Era:	
	Architects in Software Optimization	531
	<i>D. Tavagnacco, M. Frailis, S. Galeotta, E. Romelli, D. Maino, C. Vuerli,</i> <i>G. Maggio, and G. Taffoni</i>	
	GWCS - A General Approach to Astronomical World Coordinates	535
	<i>N. Dencheva and P. Greenfield</i>	
	Data-Driven Pixelation with Voronoi Tessellation	539
	<i>M. Lam and P. McWhirter</i>	
	The JWST Data Calibration Pipeline	543
	<i>H. Bushouse, J. Eisenhamer, and J. Davies</i>	
	Jitter and Readout Sampling Frequency Impact on the Athena/X-IFU Performance	547
	<i>M.T. Ceballos, B. Cobo, and P. Peille</i>	
	A Simple Survey of Cross-Matching Methods	551
	<i>D. Fan, Y. Xu, J. Han, C. Li, B. He, Y. Tao, and C. Cui</i>	
	Extragalactic Stellar Photometry and the Blending Problem	555
	<i>C. Feinstein, G. Baume, J. Rodríguez, and M. Vergne</i>	
	Astrophysical Code Migration into Exascale Era	559
	<i>D. Goz, S. Bertocco, L. Tornatore, and G. Taffoni</i>	
	Pixel Mask Filtering of the CIAO Data Model	563
	<i>H. He, M. Cresitello-Dittmar, and K. Glotfelty</i>	
	The Algorithms Behind the HPF and NEID Pipeline	567
	<i>K. F. Kaplan, C. F. Bender, R. C. Terrien, J. Ninan, A. Roy, and S. Mahadevan</i>	
	Alpha-X: An Alpha Shape-based Hierarchical Clustering Algorithm	571
	<i>R. L. Karim, L. G. Mundy, and I. Joncour</i>	
	Acceleration of the Sparse Modeling Imaging Tool for ALMA Radio Interferometric Data	575
	<i>G. Kosugi, T. Nakazato, and S. Ikeda</i>	

Tensor Clusters for Extracting and Summarizing Components in Spectral Cubes	579
<i>M. Solar, H. Farias, and C. Nunez</i>	
Robust Registration of Astronomy Catalogs	583
<i>F. Tian, T. Budavári, and A. Basu</i>	
Development of Auto-multithresh: an Automated Masking Algorithm for Deconvolution in CASA	587
<i>T. Tsutsumi, A. Kepley, I. Yoon, and U. Rau</i>	

Session XIII. Miscellaneous

Receiving Credit for Research Software	593
<i>A. Allen</i>	
Starting Up a Data Model for Exoplanetary Data	597
<i>M. Molinaro, E. Alei, S. Benatti, A. Bignamini, F. Bonnarel, M. Damasso, M. Louys, M. Maris, and V. Nascimbeni</i>	
Subaru Telescope Network 5 or STN5 - The New Computer and Network System at the Subaru Telescope	601
<i>J. Noumaru, T. Winegar, E. Kyono, H. Yamanoi, and K. Schubert</i>	
Data Products from the Europa Imaging System (EIS) on Europa Clipper	605
<i>G. W. Patterson, A. S. McEwen, E. P. Turtle, C. M. Ernst, and R. L. Kirk</i>	
The CDS HEALPix Library	609
<i>F.-X. Pineau and P. Fernique</i>	
Availability of Hyperlinked Resources in Astrophysics Papers	613
<i>P. Ryan, A. Allen, and P. J. Teuben</i>	
Data Processing of the Stratospheric Terahertz Observatory-2 [CII] Survey	617
<i>R. Shipman, Y. Seo, V. Tolls, W. Peters, Ü. Kavak, C. Kulesa, and C. Walker</i>	
VO Service in Japan: Registry Service Based on Apache Solr and SIA v2 Service for Japanese Facilities	621
<i>Y. Shirasaki, C. Zapart, M. Ohishi, and Y. Mizumoto</i>	
Running the Fermi Science Tools on Windows	625
<i>T. Stephens</i>	
Binospes@MMT: A Database Driven Model of Operations, from Planning of Observations to Data Reduction and Archiving	629
<i>I. Chilingarian, S. Moran, M. Paegert, D. Fabricant, J. Kanský, and W. Brown</i>	
QAC: Quick Array Combinations with CASA	633
<i>P. J. Teuben</i>	
Super-resolution Imaging of the Protoplanetary Disk HD 142527 using Sparse Modeling	637
<i>M. Yamaguchi, K. Akiyama, A. Kataoka, T. Tsukagoshi, T. Muto, S. Ikeda, M. Fukagawa, M. Honma, and K. Ryohei</i>	

xviii	<i>Contents</i>	
MIRISim: The JWST-MIRI Simulator		641
	<i>V. C. Geers, P. K. Klaassen, and S. Beard</i>	
Preparing for JWST: A Detailed Simulation of a MOS Deep Field with NIRSpec		645
	<i>G. Giardino, P. Ferruit, J. Chevallard, E. Curtis-Lake, N. Bonaventura, P. Jakobsen, A. Jarno, A. Pecontal, and L. Piqueras</i>	
MegaPipe 2.0: 10000 Square Degrees of CFHT MegaCam Imaging		649
	<i>S. D. J. Gwyn</i>	
Abstracting the Storage and Retrieval of Image Data at the LSST		653
	<i>T. Jenness, J. Bosch, P. Schellart, K.-T. Lim, A. Salnikov, and M. Gower</i>	
VO for Everyone - Getting Ready for the 4 th ASTERICS DADI VO School . . .		657
	<i>K. A. Lutz, M. G. Allen, A. Nebot, and S. Derriere</i>	
ALiX: An Advanced Search Interface for Aladin Lite		661
	<i>L. Michel, T. Boch, X. Shan, and J. Wang</i>	
 Session XIV. Tutorials		
All-Sky Astronomy with HiPS and MOCs		667
	<i>S. Derriere</i>	
Working with the Hubble Space Telescope Public Data on Amazon Web Services		671
	<i>I. Momcheva</i>	
 Session XV. Focus Talks and Demo Booths		
Building LOFAR as a Service		677
	<i>A. P. Mechev, J. B. R. Oonk, A. Plaat, A. Danezi, and T. W. Shimwell</i>	
Visualization in IRSA Services using Firefly		681
	<i>E. Joliet and X. Wu</i>	
Image Processing in Python with Montage		685
	<i>J. C. Good and G. B. Berriman</i>	
Workflows using Pegasus: Enabling Dark Energy Survey Pipelines		689
	<i>K. Vahi, M. Wang, C. Chang, S. Dodelson, M. Rynge, and E. Deelman</i>	
AAS Journals: Software and Data		693
	<i>F. X. Timmes and A. A. Muench</i>	

Session XVI. Birds of a Feather

Open Source/Development Software Projects and Large Organizations/Missions:
Recommendations and Challenges 697
E. Tollerud and S. Crawford

Data Formats BoF 701
J. Mink, R. I. Diaz, K. Shortridge, and T. Jenness

Data Analysis Challenges for Multi-Messenger Astrophysics 705
*P. S. Shawhan, P. R. Brady, A. Brazier, S. B. Cenko, M. Juric, and
E. Katsavounidis*

Beginners Guide to Machine Learning in Astronomy 709
K. Polsterer and N. Gianniotis

Data Citation: from Archives to Science Platforms 713
A. A. Muench and R. D’Abrusco

Unconference BoF Session: I Want to Talk About... 717
A. Allen

Session XVII. Ancillary Meetings

The ADASS Time Domain Astronomy Hackathon 723
B. A. Thomas, A. Allen, M. W. Pound, and P. J. Teuben

The International Virtual Observatory Alliance in 2018 729
M. G. Allen, P. Dowler, J. D. Evans, C. Cui, and T. Jenness

Author Index 737

Subject Index 745

ASCL Index 752

Preface

This volume of the Astronomical Society of the Pacific (ASP) Conference Series contains papers that were presented at the 28th annual conference on Astronomical Data Analysis Software and Systems (ADASS XXVIII), which was held at The Hotel at the University of Maryland, in College Park, Maryland, USA, on 11–15 November 2018. The ADASS XXVIII conference was hosted by the Astronomy Department at the University of Maryland.

1. Conference Overview

The 2018 College Park conference took place on a traditional US university campus, which for ADASS conferences is quite rare. Twice before, in 1994 and 2002, the state of Maryland saw an ADASS conference. Both of those were in Baltimore, hosted by the Space Telescope Science Institute. Given the large concentration of astronomers in the southern parts of the Baltimore-Washington area, it seemed natural for us to organize an ADASS closer to Washington DC. The Local Organizing Committee (LOC) reflected the diversity in the Baltimore-Washington area by including astronomers from STScI, NASA Goddard Space Flight Center, NASA HQ, and the University of Maryland.

The University of Maryland was founded as an agricultural college in 1856 and became a “land grant institution” in 1864. The Astronomy Department is part of the College of Computer, Mathematical, and Natural Sciences (CMNS), the largest of the twelve Schools and Colleges that constitute the University. Astronomy at Maryland was started by Uco van Wijk in 1961, who brought on Gert Westerhout a year later to head the program. The Astronomy Department was originally known as the Astronomy Program, as a unit under the Physics Department. Astronomy became a fully independent Department in 1991.

The venue for ADASS XXVIII, The Hotel, is a recently opened hotel and conference center in the newly revived College Park, with many nearby restaurants. The banquet took place at the top (penthouse) level of The Hotel, from which the top 20% of the Washington Monument (prominently featured on our conference poster) was visible! The view from the top towards the west features the traffic circle with the well known Maryland “M”, but this year will be the last time for this particular view: the new Metro Purple Line will cut through the location, forcing the “M” to be moved slightly south of its current location. Another nearby attraction is the historic College Park airport. It is the oldest continually operating airport in the world, where the Wright brothers taught the US Army how to fly in 1909.

The main business of the conference started on Monday morning with an invited talk by the CMNS Dean Amitabh Varshney introducing us to how Astronomy has inspired his field of Visual Computing. This stimulating presentation can be found online at <http://adass2018.umd.edu/abstracts/I3-1.pdf>. The conference attracted about 290 scientists from 19 different countries. A large contingent of 24 scientists applied from China, but due to not well understood delays in their visa applications, only 6 could make it to ADASS. Of the 197 presentations, 81% are published in this volume.



The Maryland “M” as seen from The Hotel (Photo: Peter Teuben)

The twelve key themes for ADASS XXVIII were: 1) Machine Learning in Astronomy; 2) Management of Large Science Projects; 3) Astrophysical Data Visualization from Line Plots to Augmented and Virtual Reality; 4) Data Science: Workflows, Hardware, Software, Humanware; 5) Science Platforms: Tools for Data Discovery and Analysis from Different Angles; 6) DevOps Practices in Astronomy Software; 7) Software for Solar System Astronomy; 8) Time Domain Astronomy; 9) Multi-Messenger Astronomy; 10) Databases and Archives: Challenges and Solutions in the Big Data Era; 11) Quality Assurance of Science Data; 12) Algorithms; and a final catch-all theme Miscellaneous. There were nearly 60 oral presentations (of which 13 were invited), 158 posters, 5 focus demonstrations, 7 Birds of a Feather sessions, 11 demo booths, 4 tutorials, and of course, one Hackathon. We should add that a record 11 tutorials were submitted in the first proposal round, which took us to a final proposal round of 6, from which the POC selected 4, presented the Sunday afternoon before the ADASS main conference.

Unlike recent years, the IVOA meeting took place before ADASS, due to a major UMD football game affecting hotel room availability. But this gave us a chance to introduce a hackathon to ADASS, during the weekend between IVOA and ADASS, hosted by the Department of Astronomy in the high tech Physical Sciences Complex building. The hackathon, a first in ADASS history, invited the community (both students and “professional” hackers) and professional astronomers to come up with a hack in the theme of Time Domain Astronomy. Generously sponsored by the City of College Park, we awarded \$1000 in prize money. Seven teams were organized, and the winning team presented *Music of Lightcurves*¹, a sonification of variable stars from the Gaia catalogue, which was received with resounding applause and a mention in the international press. The winning team combined CS/Astronomy undergraduate Siddha Mavuram, with two ADASS participants, Thomas Boch and Matthieu Baumann of the Strasbourg Astronomical Data Center (France). A report of both the IVOA meeting and the Hackathon are included near the end of these proceedings.

¹<https://github.com/tboch/lightcurves-music>

ADASS 2018 also experienced a freak snow storm on the last day of the conference, likely the first ADASS with fresh snow!

2. Organizing Committees and Sponsors

At the time of ADASS XXVIII, the semi-permanent ADASS Program Organizing Committee (POC) consisted of: Nuria Lorente (POC Chair) (AAO), Alice Allen (ASCL / UMD), Christophe Arviset (ESA-ESAC), Pascal Ballester (ESO), Sebastien Derriere (CDS / France), Kimberly DuPrie (STScI), Mike Fitzpatrick (NOAO), Stephen Gwyn (CADC), Jorge Ibsen (ALMA), Kathleen Labrie (Gemini), Mark Lacy (NRAO), Jim Lewis (IoA), Jessica Mink (SAO), Fabio Pasian (INAF), Roberto Pizzo (ASTRON), Keith Shortridge (K&V), Tadafumi Takata (NAOJ), Peter Teuben (UMD), and Xiuqin Wu (IPAC).

The Local Organizing Committee (LOC) was chaired by Peter Teuben, with other members Alice Allen (ASCL / UMD), Gerbs Bauer (UMD), Kimberly DuPrie (STScI), Tom McGlynn (IVOA / NASA Goddard), Marc Pound (UMD), Anne Raugh (UMD), Brian Thomas (NASA HQ), Elizabeth Warner (UMD), Mark Wolfire (UMD), Kevin Rauch (UMD), and Alyssa Pagan (UMD). The LOC was expertly assisted by Lisa Press and Kelly Marie Hedgepeth of UMD Conference and Visitor Services

The LOC wishes to thank CMNS Dean Amitabh Varshney and Prof. Andy Harris, chair of the Astronomy Department, for their support, and our undergraduate and graduate student volunteers for their help in making this ADASS a success. We also thank the City of College Park and our local Vigilante Coffee for sponsoring the Hackathon with cash and coffee, respectively.

ADASS XXVIII deeply appreciates the sponsorship it received from the host organization, UMD, and also from the American Astronomical Society (AAS), the North American Virtual Observatory (NAVO), Data IKU, ASCL, Teunix, LSST, Planetary Data System (PDS), Nature Astronomy, the City of College Park and Vigilante Coffee and our institutional sponsors: European Space Agency (ESA), European Southern Observatory (ESO), National Optical Astronomy Observatory (NOAO), Space Telescope Science Institute (STScI), Australian Astronomical Observatory (AAO), Smithsonian Astrophysical Observatory (SAO), Centre de Donnees Astronomiques de Strasbourg (CDS), National Astronomical Observatory of Japan (NAOJ), Infrared Processing and Analysis Center (IPAC), and Istituto Nazionale di Astrofisica (INAF). These contributions have helped keep the cost down to a manageable level for the attendees.

These proceedings contain 161 papers representing the invited, contributed, and poster papers presented at the conference as well as “Birds of a Feather” sessions, demonstrations, and reports of the IVOA activities and of ADASS’ first Hackathon. Publications of astronomical data increasingly make use of the dimension of color. Most of the figures in these proceedings are best viewed in a format that supports color. Readers are therefore encouraged to access the on-line version, which is available through the ADASS web site, <http://www.adass.org>. As has become practice now, these proceedings also include an “object index” for the Astrophysical Source Code Library (ASCL), a free on-line registry of astronomy software source code of the codes used or mentioned in the papers.

The production of these proceedings have been streamlined thanks in large part to the scripts written by Keith Shortridge. Our contribution this year enhanced this with a new workflow that with a bit more integration into the registration process will allow

even easier proceedings production for the next crew. We plan to keep this updated on github².

As for the future of ADASS, if you read the last Birds of a Feather session (A. Allen), there is a section on *Improving ADASS* suggesting we can look forward to many more years of exciting and novel conferencing!

3. Further Information

ADASS XXIX will be held in Groningen, the Netherlands, 6-10 October, 2019. Further details of this and subsequent ADASS conferences can be found at <http://www.adass.org>

Peter J. Teuben Marc W. Pound
Brian A. Thomas Elizabeth M. Warner
– The ADASS XXVIII Proceedings editors, March 2019.



The Hotel at the University of Maryland, seen from across the recreation fields that lie next to the main entrance (left tree) of campus (Photo: Peter Teuben)

More conference photos: <https://photos.app.goo.gl/yKrF7moA4vzkiAUt8>

²<https://github.com/astroumd/ADASSProceedings>

Participants

ALBERTO ACCOMAZZI, ADS / Center for Astrophysics, United States
KAREL ADÁMEK, University of Oxford, Department of Engineering Science, OeRC,
United Kingdom
MAHMUDA AFRIN BADHAN, University of Maryland College Park, United States
(volunteer)
KINGA ALBERT, Max Planck Institute for Solar System Research, Germany
FABIEN ALESINA, University of Geneva, Switzerland
ANASTASIA ALEXOV, Space Telescope Science Institute, United States
ALICE ALLEN, Astrophysics Source Code Library, United States
CHRISTOPHER ALLEN, Smithsonian Astrophysical Observatory, United States
MARK ALLEN, CNRS / Observatoire astronomique de Strasbourg, France
ANNA ALONSO, Dataiku, United States
JASPER ANNYAS, ASTRON, Netherlands
MEGAN ANSDELL, UC Berkeley, United States
MAURICIO ARAYA, UTFSM, Chile
KEITH ARNAUD, CRESST/UMD/GSFC, United States
CHRISTOPHE ARVISET, ESA-ESAC, Spain
JOSEPH ASERCION, NASA Goddard Space Flight Center, United States
CARLO BAFFA, INAF - OAA, Italy
DEBORAH BAINES, ESA - ESAC, Spain
JORGO BAKKER, ESAC/ESA, Spain
PASCAL BALLESTER, ESO, Germany
JAMES BAUER, University of Maryland, United States
MATTHIEU BAUMANN, CNRS, Observatoire de Strasbourg, France
UGO BECCIANI, Istituto Nazionale di Astrofisica, Italy
STEFAN BECKER, Universität Heidelberg, Germany
BRUCE BERRIMAN, Caltech/IPAC-NExScI, United States
SERGI BLANCO-CUARESMA, Harvard-Smithsonian Center for Astrophysics, United States
THOMAS BOCH, CNRS, Observatoire de Strasbourg, France
CATHERINE BOISSON, LUTH - Observatoire de Paris, France
FRANÇOIS BONNAREL, CDS ObAS, France
JOSEPH BOOKER, Johns Hopkins University, United States
KIRK BORNE, BOOZ Allen Hamilton, United States
JAMES BOSCH, Princeton University / LSST, United States
MALIK OLIVIER BOUSSEJRA, Keio University, Japan
CLARA BRASSEUR, Space Telescope Science Institute, United States

xxvi

Participants

ADAM BRAZIER, Cornell University, United States
MATTHEW BROWN, W. M. Keck Observatory, United States
NICOLAS BUCHSCHACHER, University of Geneva, Switzerland
JULIEN BURNIER, Université de Genève, Switzerland
HOWARD BUSHOUSE, STScI, United States
IVO BUSKO, Space Telescope Science Institute, United States
NICOLA CAON, Instituto de Astrofísica de Canarias, Spain
NICOLÁS CARDIEL, Universidad Complutense de Madrid, Spain
M.TERESA CEBALLOS, Instituto de Física de Cantabria (CSIC-UC), Spain
IGOR CHILINGARIAN, SAO, United States
SELINA CHU, NASA Jet Propulsion Laboratory, United States
ANGUS COMRIE, University of Cape Town, South Africa
VALERIE CONNAUGHTON, NASA HQ, United States
MICHAEL CORCORAN, The Catholic University of America, United States
ALAIN COULAIS, Observatoire de Paris (LERMA), France
STEVEN CRAWFORD, Space Telescope Science Institute, United States
MARK CRESITELLO-DITTMAR, Smithsonian Astrophysical Observatory, United States
ANDRE CSILLAGHY, FHNW, Switzerland
CHENZHOU CUI, NAOC, China
DANIEL DA SILVA, NASA/GSFC, United States
CHRISTOPHE DABIN, CNES, France
RAFFAELE D'ABRUSCO, CXC/SAO, United States
CONG DAI, Beijing Normal University, China
PHILIP DALY, Steward Observatory, United States
ARANCHIA DELGADO, Institute of Astronomy, University of Cambridge, United Kingdom
NADIA DENCHEVA, Space Telescope Science Institute, United States
VLADIMIR DERGACHEV, Albert Einstein Institute Hannover, Germany
SÉBASTIEN DERRIERE, CDS, Université de Strasbourg, France
VANDANA DESAI, Caltech / IPAC, United States
SHUBHANKAR DESHPANDE, Carnegie Mellon University, United States
ERIK DEUL, Leiden Observatory, Netherlands
ROSA DIAZ, STScI, United States
TOM DONALDSON, Space Telescope Science Institute, United States
THERESA DOWER, STScI, United States
PATRICK DOWLER, National Research Council Canada, Canada
CHRISTOPH DREISSIGACKER, Albert Einstein Institute Hannover, Germany
RICHARD DUBOIS, SLAC National Accelerator Laboratory, United States

Participants

xxvii

GREGORY DUBOIS-FELSMANN, Caltech/IPAC-LSST, United States
KIMBERLY DUPRIE, STScI, United States
DANIEL DURAND, National Research Council Canada - CADC, Canada
RICK EBERT, Caltech IPAC, United States
KEN EBISAWA, ISAS/JAXA, Japan
JOSEPH EGGEN, University of Maryland, United States
SATOSHI EGUCHI, Fukuoka University, Japan
JONATHAN EISENHAMER, AURA/STScI, United States
BJORN EMONTS, NRAO, United States
JANET EVANS, Harvard Smithsonian CfA, United States
DONGWEI FAN, National Astronomical Observatories, CAS, China
CARLOS FEINSTEIN, Observatorio Astronomico, Argentina
HENRY FERGUSON, Space Telescope Science Institute, United States
PIERRE FERNIQUE, CDS, University of Strasbourg, France
MICHAEL FITZPATRICK, NOAO, United States
MATTHIAS FUESSLING, CTA Observatory, Germany
CARLOS GABRIEL, ESA / ESAC, Spain
KINGSLEY GALE-SIDES, University of Cambridge, United Kingdom
SÉVERIN GAUDET, NRC/HAA/CADC, Canada
VINCENT GEERS, UK Astronomy Technology Centre / STFC / UKRI, United Kingdom
GIOVANNA GIARDINO, ESA/ESTEC, Netherlands
SANKALP GILDA, University of Florida, United States
RANPAL GILL, LSST/AURA, United States
JUAN GONZALEZ-NUÑEZ, European Space Agency, Spain
JOHN GOOD, Caltech/IPAC-NExScI, United States
CRAIG GORDON, NASA/Goddard Space Flight Center, Innovim, United States
DAVID GOZ, INAF - OATs, Italy
GONZALO GRACIA ABRIL, Gaia DPAC Project Office, ESAC-ESA and Astronomisches
Rechen-Institut, ZAH, Universität Heidelberg, Spain
YAN GRANGE, ASTRON, Netherlands
KIRILL GRISHIN, Moscow State University, Russian Federation
STEVE GROOM, Caltech / IPAC, United States
DAVID GRUMM, STScI, United States
LUIZ FERNANDO GUEDES DOS SANTOS, The Catholic University of America, United States
ROCIO GUERRA NOGUERO, ESA, Spain
PRAMOD GUPTA, University of Washington, United States
STEPHEN GWYN, Canadian Astronomy Data Centre, Canada

xxviii

Participants

JONATHAN HARGIS, STScI, United States

ADAM HARVEY, University of Maryland, Baltimore County, United States

HELEN HE, SAO, United States

HENDRIK HEINL, ARI / GAVO, Germany

RONALD HENRY, Space Telescope Science Institute, United States

FABIO HERNANDEZ, CC-IN2P3, France

MATTHEW HOLMAN, Harvard-Smithsonian Center for Astrophysics, United States

DONALD HORNER, NASA Goddard Spaceflight Center, United States

MINH HUYN, CSIRO / ICRAR, Australia

JORGE IBSEN, Joint ALMA Observatory (ESO), Chile

SHIRO IKEDA, The Institute of Statistical Mathematics, Japan

BRYAN IRBY, NASA Goddard Space Flight Center, United States

JACK IRELAND, NASA Goddard Spaceflight Center, United States

HIROYOSHI IWASAKI, Rikkyo University, Japan

TESS JAFFE, U. of Maryland and NASA/GSFC, United States

TIM JENNESS, LSST/AURA, United States

ERIC JESCHKE, National Astronomical Observatory of Japan, United States

KYLE JOHNSTON, Florida Institute of Technology, United States

EMMANUEL JOLIET, IPAC/Caltech, United States

ISABELLE JONCOUR, UGA/UMD, France

MARIO JURIC, University of Washington, United States

CATHERINE KALEIDA, The Space Telescope Science Institute, United States

KYLE KAPLAN, The University of Arizona, United States

RAMSEY KARIM, University of Maryland, College Park, United States (volunteer)

ZEKE KAUFMAN, Smithsonian Astrophysical Observatory, United States

WATARU KAWASAKI, NAOJ, Japan

MICHAEL KELLEY, University of Maryland, United States

BRIAN KENT, NRAO, United States

AMANDA KEPLEY, National Radio Astronomy Observatory, United States

VYACHESLAV KITAEFF, ICRAR-CSIRO, Australia

AUKE KLAZEMA, Astron, Netherlands

XIAO KONG, National Astronomical Observatories, CAS, China

KARL KOSACK, CEA Paris-Saclay, France

GEORGE KOSUGI, National Astronomical Observatory of Japan, Japan

ERIK KUULKERS, European Space Agency, Netherlands

KATHLEEN LABRIE, Gemini Observatory, United States

MARK LACY, NRAO, United States

Participants

xxix

CHEUK YIN LAM, Liverpool Telescope, United Kingdom
JASON LAMMERS, University of Maryland College Park, United States (volunteer)
GILLES LANDAIS, Observatoire Astronomique de Strasbourg (CDS), France
MARCO LANDONI, Istituto Nazionale di Astrofisica, Italy
OMAR LAURINO, Smithsonian Astrophysical Observatory, United States
JAMES LEWIS, University of Cambridge, United Kingdom
MAGGIE LIEU, ESA, Spain
PEY LIAN LIM, STScI, United States
KENNY LO, LSST / SLAC, United States
TAK LO, Caltech/IPAC, United States
MIN LONG, Boise State University, United States
MARCEL LOOSE, ASTRON, Netherlands
NURIA LORENTE, AAO - MQ, Australia
MIREILLE LOUYS, Icube, CDS, University of Strasbourg, France
YUXI LU, Univeristy of Maryland, United States (volunteer)
MICHAEL LUNDQUIST, University of Arizona/Steward Observatory, United States
KATHARINA LUTZ, CDS, Observatoire astronomique de Strasbourg, France
JEFF MADER, W. M. Keck Observatory, United States
BRIAN MAJOR, CADC - National Research Council Canada, Canada
BEATRIZ MARTINEZ, Rhea Systems S.A. for ESA, Spain
PATRICK MASI-PHELPS, Dataiku, United States
THOMAS MCGLYNN, NASA/GSFC, United States
PAUL ROSS McWHIRTER, Liverpool John Moores University, United Kingdom
ALEXANDAR MECHEV, Leiden University, Netherlands
PABLO MELLADO, IRAM, Spain
LAURENT MICHEL, SSC XMM-Newton - Observatoire de Strasbourg, France
ALBERTO MICOL, ESO, Germany
JENNIFER MILLER, Gemini Observatory, United States
CHASE MILLION, Million Concepts, United States
JESSICA MINK, Smithsonian Astrophysical Observatory, United States
MARCO MOLINARO, INAF - OATs, Italy
IVELINA MOMCHEVA, STScI, United States
FEDERICO MONTESINO POUZOLS, ESO, Germany
MIKIO MORII, Institute of Statistical Mathematics, Japan
DAVE MORRIS, University of Edinburgh, United Kingdom
SKARLETH MOTINO FLORES, Catholic University of America, United States
AUGUST MUENCH, American Astronomical Society, United States

xxx

Participants

TAKESHI NAKAZATO, National Astronomical Observatory of Japan, Japan
VICENTE NAVARRO, ESA, Spain
ADA NEBOT, CDS, Observatoire Astronomique de Strasbourg, France
SARA NIETO, ESA, Spain
BOJAN NIKOLIC, University of Cambridge, United Kingdom
JUNICHI NOMARU, Subaru Telescope, United States
EDUARDO OJERO-PASCUAL, European Space Astronomy Centre, Spain
WILLIAM O'MULLANE, AURA/LSST, United States
SIMON O'TOOLE, AAO-MQ, Australia
MAXIME PAILLASSA, University of Bordeaux / CNES, France
DAVE PALLOT, University of Western Australia, Australia
JOSE PARRA, Joint ALMA Observatory (ESO), Chile
SERGIO PASCUAL, Universidad Complutense de Madrid, Spain
GERALD PATTERSON, Johns Hopkins University Applied Physics Laboratory, United States
CHARLES PAXSON, Smithsonian Astrophysical Observatory, United States
JOSE VICENTE PEREA-CALDERON, RHEA for ESA/ESAC, Spain
JOSHUA PEEK, STScI, United States
FERNANDO PEREZ, ESAC, Spain
ABIGAIL PETULANTE, Vanderbilt University, United States
TIMOTHY PICKERING, MMT Observatory, United States
FRANCOIS-XAVIER PINEAU, CNRS, Observatoire de Strasbourg, France
ROBERTO PIZZO, ASTRON, Netherlands
RAYMOND PLANTE, National Institute of Standards and Technology, United States
EMIL POLISENSKY, Naval Research Laboratory, United States
KAI POLSTERER, HITS gGmbH, Germany
MARC POUND, University of Maryland, United States
REINHARD PRIX, Albert-Einstein-Institute Hannover, Germany
RYAN RABA, NRAO, United States
ELENA RACERO, ESAC - Serco, Spain
MICHAEL RADDICK, Johns Hopkins University, United States
EMANUEL RAMIREZ, Quasar Science Resources, Spain
ANNE RAUGH, University of Maryland, United States
ROSLY RENIL, SARA0/NRF, South Africa
LUCA RIZZI, W. M. Keck Observatory, United States
ERIK ROMELLI, INAF-OATs, Italy
EVGENII RUBTSOV, MSU, Faculty of Physics, Russian Federation

Participants

xxxi

KRISTIN RUTKOWSKI, NASA / Goddard, United States
P. WESLEY RYAN, Astrophysics Source Code Library, United States
JESUS SALGADO, ESDC/ESAC/ESA, Spain
MICHELE SANGUILLON, LUPM, France
RAFAEL SANTOS, INPE, Brazil
RENAUD SAVALLE, Paris Observatory, France
ANDRE SCHAAFF, CNRS / Observatoire astronomique de Strasbourg, France
MATHIEU SERVILLAT, LUTH - Observatoire de Paris, France
NIGEL SHARP, US National Science Foundation, United States
PETER SHAWHAN, University of Maryland, United States
ROBERT SHEN, Astronomy Australia Limited, Australia
MIN-SU SHIN, Korea Astronomy and Space Science Institute, Republic of Korea
DANIEL SHIPMAN, University of Southern Maine, United States
RUSSELL SHIPMAN, SRON, Netherlands
YUJI SHIRASAKI, National Astronomical Observatory of Japan, Japan
KEITH SHORTRIDGE, K&V, Australia
ALAN SMALE, NASA, United States
ARFON SMITH, STScI, United States
JEFFREY SMITH, SETI Institute, United States
MEGAN SMITH, UMD Physics Department, United States
GREGORY SNYDER, Space Telescope Science Institute, United States
MARTIN SOLAR, Universidad Andres Bello, Chile
MAURICIO SOLAR, Federico Santa Maria Technical University, Chile
MAX SPOLAOR, NASA - TMC, United States
TOM STEPHENS, GSFC/Innovim, United States
FELIX STOEHR, ESO/ALMA, Germany
OLE STREICHER, Leibniz Institute for Astrophysics Potsdam, Germany
RAYMOND SUTRISNO, University of Houston, United States
DARYL SWADE, STScI, United States
TADAFUMI TAKATA, National Astronomical Observatory of Japan, Japan
DANIELE TAVAGNACCO, INAF-OATs, Italy
MARK TAYLOR, University of Bristol, United Kingdom
PETER TEUBEN, University of Maryland, United States
BRIAN THOMAS, NASA, United States
FAN TIAN, Johns Hopkins University, United States
FRANK TIMMES, Arizona State University, United States
IGNACIO TOLEDO, Joint ALMA Observatory, Chile

xxxii

Participants

ERIK TOLLERUD, STScI, United States

MAURIZIO TOMASI, Università degli Studi di Milano, Italy

SIMÓN TORRES, SOAR Telescope, Chile

TAKAHIRO TSUTSUMI, NRAO, United States

KARAN VAHI, Pegasus Team - USC / ISI, United States

BRIAN VAN KLAVEREN, SLAC, United States

JOSH VEITCH-MICHAELIS, Liverpool John Moores University, United Kingdom

NICO VERMAAS, Astron, Netherlands

RUI WANG, National Astronomical Observatories, CAS, China

ELIZABETH WARNER, University of Maryland, United States

ANDREAS WICENEC, University of Western Australia, Australia

PETER WILLIAMS, Harvard-Smithsonian Center for Astrophysics, United States

ERIC WINTER, Space Telescope Science Institute, United States

MICHAEL WISE, ASTRON, Netherlands

PAUL WOODS, Nature Astronomy, United Kingdom

XIUQIN WU, Caltech/IPAC-LSST, United States

ZHENYU WU, National Astronomical Observatories, CAS, China

MASAYUKI YAMAGUCHI, The University of Tokyo/NAOJ, Japan

MICHAEL YOUNG, Indiana University, United States

CHRISTOPHER ZAPART, National Astronomical Observatory of Japan, Japan

PETAR ZECEVIC, University of Zagreb, Croatia

TAYLOR BARTLOW, University of Maryland, United States (volunteer)

EVAN DAVIS, University of Maryland, United States (volunteer)

BLAKE HARTLEY, University of Maryland, United States (volunteer)

JEGUG IH, University of Maryland, United States (volunteer)

ALYSSA PAGAN, University of Maryland, United States (volunteer)

LUCIA PEREZ, University of Maryland, United States (volunteer)

KEVIN RAUCH, University of Maryland, United States (volunteer)

NAYLYNN TANON REYES, University of Maryland, United States (volunteer)

LIZ TARANTINO, University of Maryland, United States (volunteer)

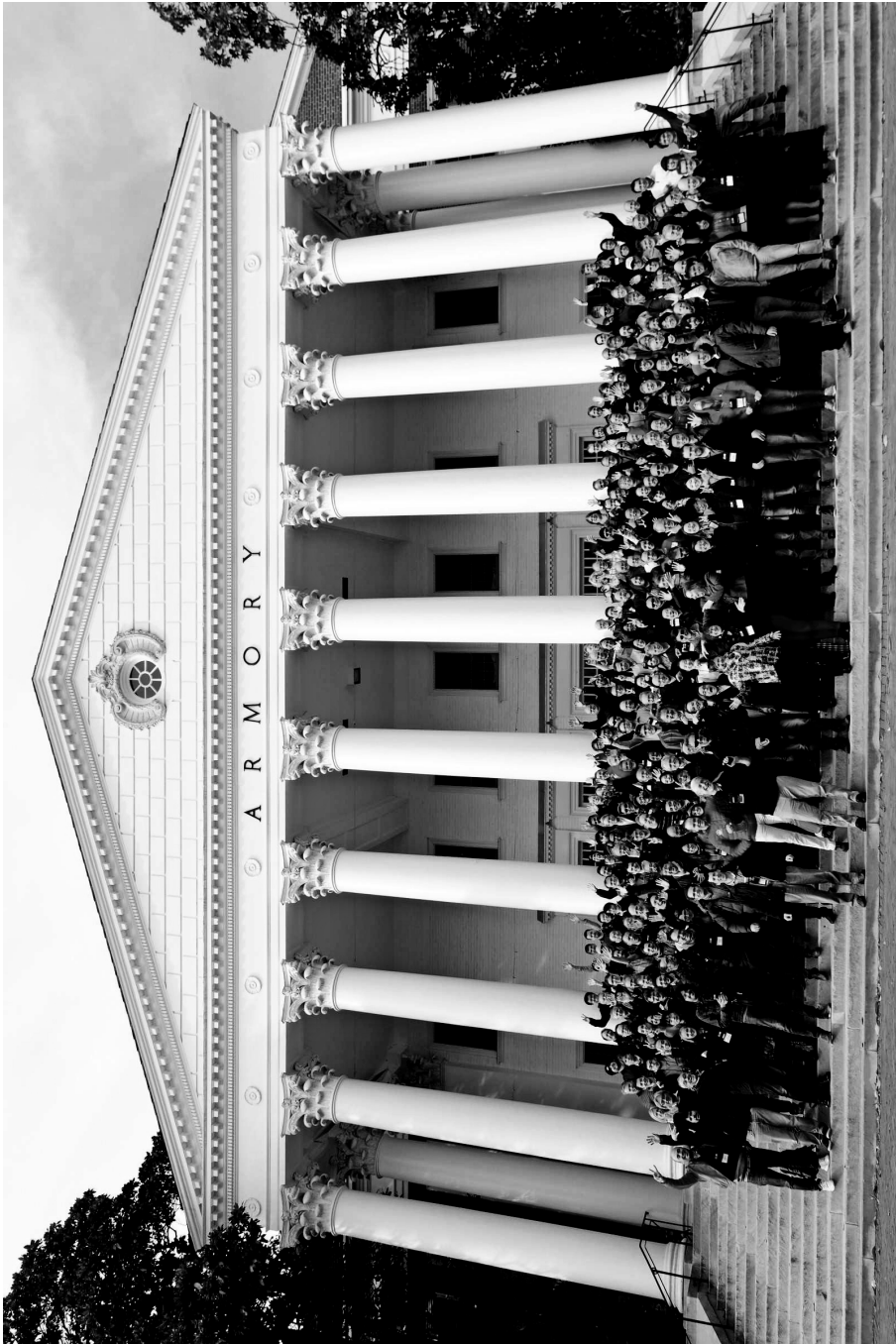
MIGUEL ANGEL REQUENA TORRES, University of Maryland, United States (volunteer)

CHARLOTTE WARD, University of Maryland, United States (volunteer)

DMITRIY YARUNIN, University of Maryland, United States (volunteer)

LISA PRESS, University of Maryland Conference Services, United States

KELLY MARIE HEDGEPEETH, University of Maryland Conference Services, United States



ADASS 2018 participants in front of The Armory (Photo: Lisa Press).

Session I

Astrophysical Data Visualization from Line Plots to Augmented and Virtual Reality

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

3D Data Visualization in Astrophysics

Brian R. Kent

National Radio Astronomy Observatory
 520 Edgemont Road, Charlottesville, VA, USA; bkent@nrao.edu

Abstract. We present unique methods for rendering astronomical data - 3D galaxy catalogs, planetary maps, data cubes, and simulations. Using tools and languages including Blender, Python, and Google Spatial Media, a user can render their own science results, allowing for further analysis of their data phase space. We aim to put these tools and methods in the hands of students and researchers so that they can bring their own data visualizations to life on different computing platforms.

<https://www.cv.nrao.edu/~bkent/blender/>

1. Introduction

Data visualization is a critical component of astronomical research. Large and complex data need innovative methods for display and analysis. Databases contain catalog surveys with hundreds of parameters creating large phase spaces to be explored. In addition, astronomical data provides some of the most inspiring and visually stunning images and simulations that the scientific community has to offer. Research tools for both scientists and broader impact to the intrigued public requires useful software tools to facilitate visualizing data. Tools are built by innovators within the astronomical community, others are adapted from software in other fields. The cross-disciplinary research creates beneficial resource and knowledge sharing between astronomy, 3D graphics, and data sciences (Kent 2017a).

Visualization using different methods can give insight into astronomical data. It is this data exploration that can shed new light on and lead to new discoveries with imaging, maps, catalogs and multidimensional data cubes. With the the increase in archival data products, research can benefit from *re-visualizing* prior epochs of data in new ways (Berriman & Groom 2011). Figure 1 shows the increase in archive, survey, and project volume vs. time. Visualization techniques and tools have become critical elements of astronomical research in the era of large surveys and high data rates.

Phase spaces of data sets with $N > 2$ require either a reduction in the number of dimensions to a two-dimensional plot or 3D rendering. With 3D graphics, one can see more information about a given set of data in one view, move in, around, and through a visualization. If there is a time series associated with an observation or simulation, it can be animated and rendered from multiple camera viewpoints. Immersive data experiences (virtual reality/augmented reality/cave wall) allow a user to explore data with devices they always have on their person - namely a mobile phone or tablet. The accelerometer hardware present in said devices can put a user *inside* the data and allow

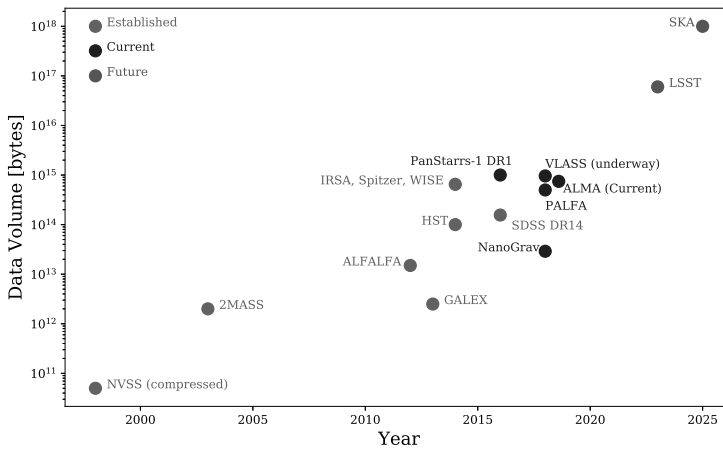


Figure 1. Increasing data volumes from established, current, and future astronomical surveys and observatories. References for these data points are as follows.

NVSS: <https://www.cv.nrao.edu/nvss/>

2MASS: <https://old.ipac.caltech.edu/2mass/overview/>

ALFALFA: Haynes et al. (2018)

GALEX: Bianchi (2014)

HST: <https://registry.opendata.aws/hst/>

IRSA, Spitzer, WISE: Berriman & Groom (2011)

SDSS: https://www.sdss.org/dr14/data_access/

PanStarrs: <https://panstarrs.stsci.edu/>

VLASS: Myers et al. (2015)

NanoGrav and PALFA: Demorest & Brazier (2018)

ALMA: Lacy & Halstead (2015)

LSST: <https://www.lsst.org/about/dm>

SKA: <https://www.skatelescope.org/>

visual data inspection and discovery (Kent 2017b). Whatever the application or goal, a 3D rendering can often enhance a data visualization scenario.

In these proceedings we specifically review a brief history of 3D data visualization, tools and types of methods in astronomy, software in the 3D graphics industry, Blender, a Python-API based 3D rendering software package, and Blender's usage in astronomy and astrophysics.

2. History

From a certain point of view, science has always relied on some form of data presentation or visualization to convey the results of an observation or experiment. Astronomy has a long history dating to antiquity of charting the heavens, and tabulating and graphing their temporal motions. Imaging ranging from plates and film to digital allowed astronomers to record and preserve what was detected at a particular time and vantage

point. Radio receivers and high energy detectors expand our EM view of the Universe and push the boundaries of time-domain astronomy and the speed at which we can respond to target-of-opportunity events.

Two-dimensional plots act as a standard display to identify trends among data variable. Higher-dimensional data sets must either reduce the number of dimensions, use a 3D display, color with transparency accordingly, or use a combination of all three. Exploratory analysis and visualization can give insight into N-dimensional data with linked views – allowing a scientist to visually mine their data and any statistical properties (Goodman 2012). The availability of a wide variety of data and metadata parameters leads to science-driven development (Fitzpatrick et al. 2016; Graham et al. 2016).

Data visualization has now extended beyond the pages of journals and our desktop screens to virtual and augmented reality (VR/AR) and the mobile devices and tablets ever present in our hands (Vogt & Shingles 2013). These immersive data applications can put the user *in* their data space - while they have a certain aesthetic appeal and definite education and public outreach applications, they also can be used for research. Applications include collaborative visual analytics (Vohl et al. 2017b), multi-screen CAVE viewing (Vohl et al. 2017a), 3D printing (Madura 2017), and interactive applications (Punzo et al. 2015; Vogt et al. 2017).

3. Modules, tools, and libraries for 3D rendering

Defining software tools for data visualization can be a bit amorphous. A tool can be a full-featured astro-specific package, an ancillary library of classes and functions, or software from another technical area of research that can be adapted for use in astronomy. Tutorials written with code in a rich-text format or markup, or narrated video tutorials can train students and users on how to import and manipulate their data in new packages.

Astronomy packages like AstroPy (Astropy Collaboration et al. 2013), Kapteyn (Terlouw & Vogelaar 2016), and Montage (Berriman et al. 2007; Jacob et al. 2010) are used in conjunction with 3D graphics software, allowing a user to manipulate data in a Python environment before rendering. The flexibility of using modules makes 3D rendering packages versatile in manipulating different types of astronomical data.

Commercial packages used in the graphics industry are not traditional pieces of software used in the astronomical community. Maya¹, 3D Studio Max², and Lightwave³ are full featured software packages that can be used to render 3D data. The Pixar package Renderman⁴ can act as a backend renderer for several modeling GUIs. Others, like Houdini⁵, have been successfully adapted for use in astronomy (Naiman et al. 2017).

¹<https://www.autodesk.com/products/maya>

²<https://www.autodesk.com/products/3ds-max>

³<https://www.lightwave3d.com/>

⁴<https://renderman.pixar.com/>

⁵<https://www.sidefx.com/>

4. Blender for Astronomy

Blender is an extremely versatile 3D graphics rendering package⁶. Its Python scripting capabilities and extensive graphical user interface make it a natural fit for astronomical data reduction [Kent 2013]. The utility of the package includes 3D modeling, 2D and 3D texturing, 3D voxel rendering, animation, lighting, camera control, and node compositing. Each of these features can be used alone or in concert for various forms of astronomical data visualization (Kent 2013, 2015).

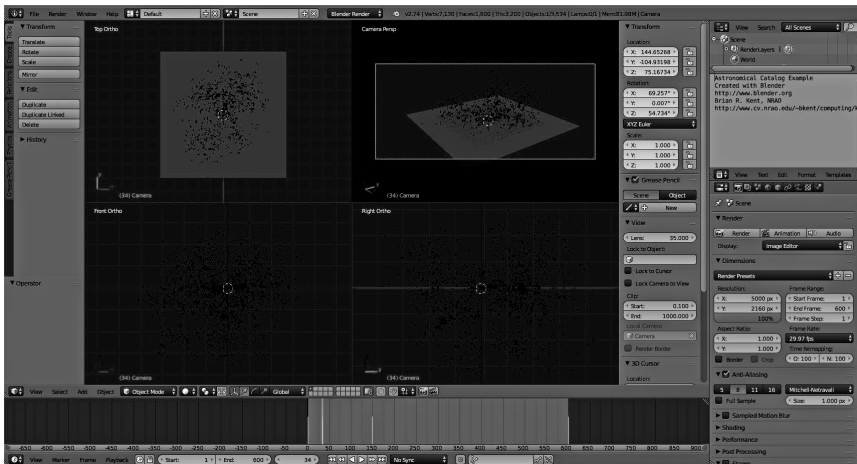


Figure 2. Interface of Blender showing a 3D data visualization of galaxies from multiple angles. The bottom interface shows markers for animation and camera key frames. This particular rendering will be used for spherical panoramic video for use on mobile and tablet devices.

- **Meshes.** Blender objects are built upon meshes - collections of vertex points, connecting lines, and faces. Meshes can be rendered as wire frames (useful for an extragalactic distance grid), as shaded polygons (useful for representative models or simulations), voxel containers (transparent data cubes), or textured 3D surfaces (planetary maps).
- **Cameras.** Rendering in Blender occurs from the view point of a Camera object. Focal length, detector size and resolution, field of view, and projection are all properties of a Blender camera object.
- **Lighting.** Lighting is accomplished via both emission and reflection mechanisms, usually with solid polygon surfaces. Wire meshes can be *self-illuminating* irregardless of where lighting elements are placed in a visualization scene.
- **Animation.** All meshes, lighting elements, and cameras can be animated. Animation involves translation, rotation, and scaling and the associated rates of those elements.

⁶<https://www.blender.org/>

Blender also has plug-ins for nVidia CUDA and OpenCL for GPU hardware acceleration in the rendering process. Depending on the application, this can greatly decrease the rendering time needed for visualization.

The Blender workflow for 3D astronomical rendering first requires the investigator to identify what kind of visualization they wish to make. Is it a single frame or animation? What platform will users view the rendering (video, mobile device/tablet)? Is it a physical model or does a 3D mesh object act as a container for a data cube? What are the requirements for the final rendering output? Is transparency or ray tracing needed? What camera angles are needed to show unique aspects of the data set for review and analysis?

4.1. Data visualization Examples

Astronomers have used Blender in a variety of ways in their work - developing code, tutorials, and examples for the community to build upon.

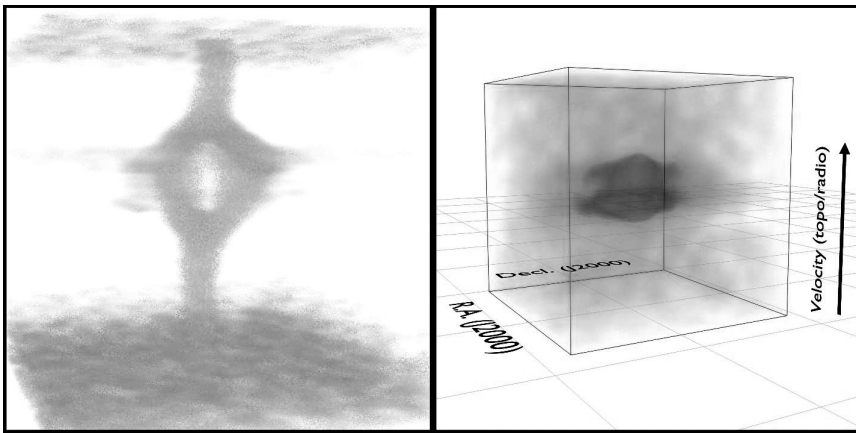


Figure 3. Data cubes with data from ALMA featuring *Left*: a protoplanetary gap (Casassus et al. 2013) and *Right*: HCN in the inner coma of comet C/2012 F6 Lemmon (Cordiner et al. 2014).

Examples are as follows:

- **Data cubes.** The voxel data structure in Blender allows a user to render data cubes transparently. FRELLED (Taylor 2015, 2017) has successfully demonstrated the concept with neutral hydrogen surveys. Gárate (2017) has used Blender to render magneto-hydrodynamic simulations. ALMA data have been successfully rendered using techniques involving Blender (Figure 3).
- **Simulations.** Figure 4 shows a snapshot from a galaxy collision simulation using data from GADGET-2 (Springel 2005). AstroBlend⁷, an open-source Python library combines Blender with yt (Turk et al. 2011; Naiman 2016).

⁷<http://www.astroblend.com/>

- **Catalogs.** Figure 5 shows a 3D galaxy catalog rendering generated from the Extragalactic Distance Database, EDD (Tully et al. 2009).
- **Surface maps.** Figure 6 shows a displacement map of Martian shield volcano Olympus Mons with data from Christensen et al. (2001). Florinsky et al. (2018) have created morphometric globes for Mars and the Moon using Blender with a web interface.
- **EPO.** Diemer & Facio (2017) has used 3D printing and textile interfaces to create museum displays of cosmological large scale structure.

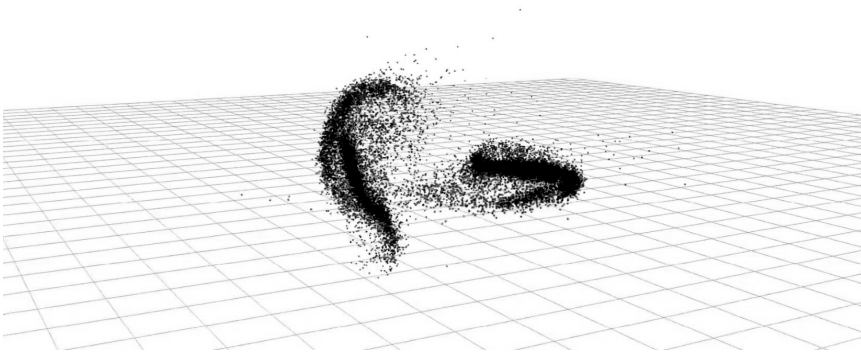


Figure 4. N-body simulation

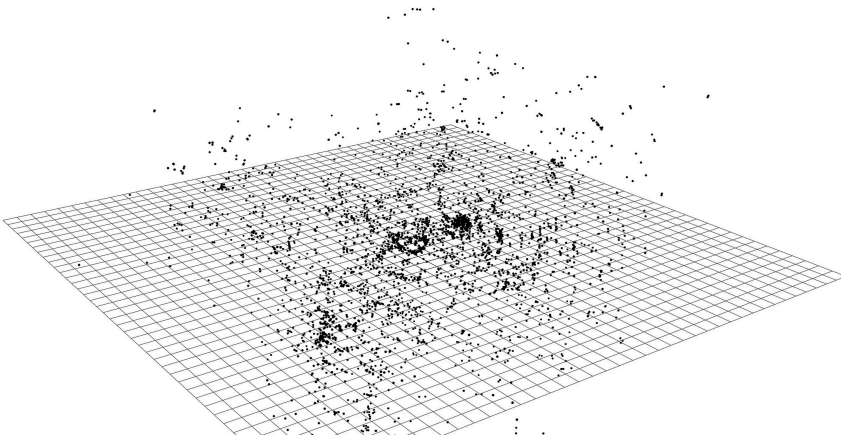


Figure 5. Extragalactic catalog rendering using data from Tully et al. (2009)

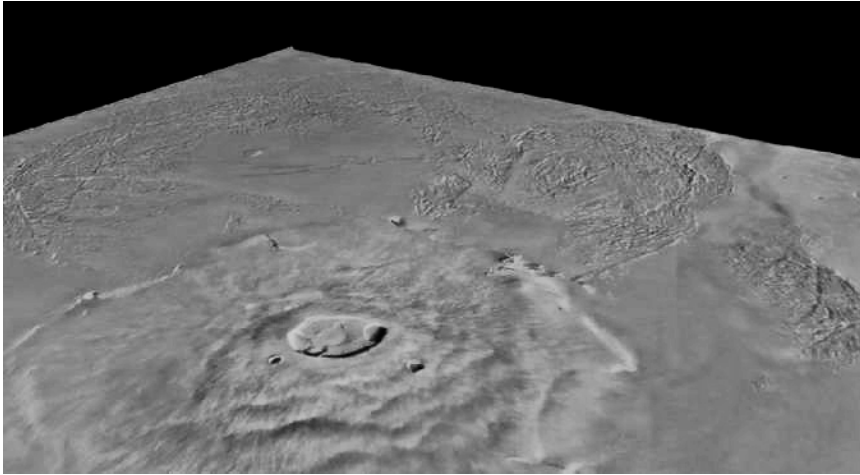


Figure 6. 3D surface rendering of Olympus Mons on Mars.

5. Mobile devices and interactivity

One of the best interactive data viewers is carried in our hands all the time. Mobile phones and tablets have high resolution displays and accelerometers that can allow a user to interactively view data - 3D models, all sky maps, or catalogs. Kent (2017b) and Fluke & Barnes (2018) detail how to do this using two different methods (Figure 7). Google's Spatial Media module⁸, available as a standalone program or in Python, can take a spherical 360 degree video, inject metadata into the video header, and then have it be ready for injection in to the video sharing website YouTube.⁹

Acknowledgments. The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

⁸<https://github.com/google/spatial-media>

⁹Examples can be found on the author's channel *Visualize Astronomy*:
<https://www.youtube.com/user/VisualizeAstronomy/>

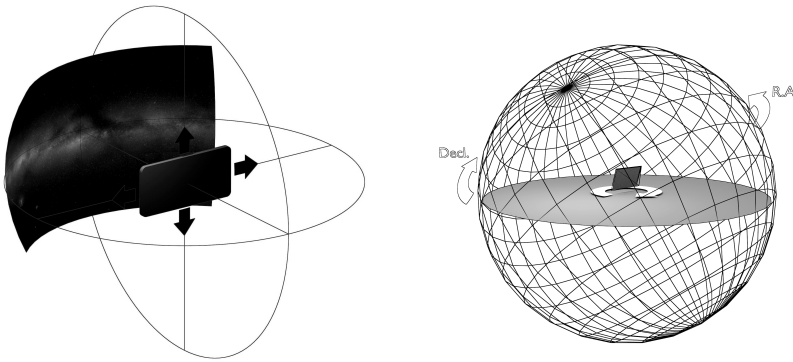


Figure 7. *Left:* A mobile phone or tablet can be used to view an all sky map. *Right:* An all sky map projection (Kent 2017b).

References

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., Unther, H. M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J. E. H., Singer, L., Fox, R., Weaver, B. A., Zabalza, V., Edwards, Z. I., Azalee Bostroem, K., Burke, D. J., Casey, A. R., Crawford, S. M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P. L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., & Streicher, O. 2013, *A&A*, 558, A33. 1307.6212
- Berriman, G. B., & Groom, S. L. 2011, *ACM Queue*, 9, 21:20. URL <http://doi.acm.org/10.1145/2039359.2047483>
- Berriman, G. B., Laity, A. C., Good, J. C., Katz, D. S., Jacob, J. C., Deelman, E., Singh, G., Su, M.-H., & Prince, T. A. 2007, *Highlights of Astronomy*, 14, 621
- Bianchi, L. 2014, *Ap&SS*, 354, 103. 1404.4882
- Casassus, S., van der Plas, G., M., S. P., Dent, W. R. F., Fomalont, E., Hagelberg, J., Hales, A., Jordán, A., Mawet, D., Ménard, F., Wootten, A., Wilner, D., Hughes, A. M., Schreiber, M. R., Girard, J. H., Ercolano, B., Canovas, H., Román, P. E., & Salinas, V. 2013, *Nature*, 493, 191. 1305.6062
- Christensen, P. R., Bandfield, J. L., Hamilton, V. E., Ruff, S. W., Kieffer, H. H., Titus, T. N., Malin, M. C., Morris, R. V., Lane, M. D., Clark, R. L., Jakosky, B. M., Mellon, M. T., Pearl, J. C., Conrath, B. J., Smith, M. D., Clancy, R. T., Kuzmin, R. O., Roush, T., Mehall, G. L., Gorelick, N., Bender, K., Murray, K., Dason, S., Greene, E., Silverman, S., & Greenfield, M. 2001, *Journal of Geophysical Research*, 106, 23823
- Cordiner, M. A., Remijan, A. J., Boissier, J., Milam, S. N., Mumma, M. J., Charnley, S. B., Paganini, L., Villanueva, G., Bockelée-Morvan, D., Kuan, Y.-J., Chuang, Y.-L., Lis, D. C., Biver, N., Crovisier, J., Minniti, D., & Coulson, I. M. 2014, *ApJ*, 792, L2
- Demorest, P., & Brazier, A. 2018, Private Communication
- Diemer, B., & Facio, I. 2017, *PASP*, 129, 058013. 1702.03897
- Fitzpatrick, M. J., Graham, M. J., Mighell, K. J., Olsen, K., Norris, P., Ridgway, S. T., Stobie, E. B., Bolton, A. S., Saha, A., & Huang, L. W. 2016, in *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913 of *Proc. of the SPIE*, 99130L
- Florinsky, I. V., Garov, A. S., & Karachevtseva, I. P. 2018, *Planetary and Space Science*, 159, 105
- Fluke, C. J., & Barnes, D. G. 2018, *PASA*, 35, e026. 1805.03354
- Gárate, M. 2017, *PASP*, 129, 058010. 1611.06965
- Goodman, A. A. 2012, *Astronomische Nachrichten*, 333, 505. 1205.4747
- Graham, M. J., Fitzpatrick, M. J., Norris, P., Mighell, K. J., Olsen, K., Stobie, E. B., Ridgway, S. T., Bolton, A. S., Saha, A., & Huang, L. W. 2016, in *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913 of *Proc. of the SPIE*, 99131I
- Haynes, M. P., Giovanelli, R., Kent, B. R., Adams, E. A. K., Balonek, T. J., Craig, D. W., Fertig, D., Finn, R., Giovanardi, C., Hallenbeck, G., Hess, K. M., Hoffman, G. L., Huang, S., Jones, M. G., Koopmann, R. A., Kornreich, D. A., Leisman, L., Miller, J., Moorman, C., O'Connor, J., O'Donoghue, A., Papastergis, E., Troischt, P., Stark, D., & Xiao, L. 2018, *ApJ*, 861, 49. 1805.11499
- Jacob, J. C., Katz, D. S., Berriman, G. B., Good, J., Laity, A. C., Deelman, E., Kesselman, C., Singh, G., Su, M.-H., Prince, T. A., & Williams, R. 2010, *Montage: An Astronomical Image Mosaicking Toolkit*, *Astrophysics Source Code Library*. 1010.036
- Kent, B. R. 2013, *PASP*, 125, 731. 1306.3481
- 2015, *3D Scientific Visualization with Blender*
- 2017a, *PASP*, 129, 058001. 1705.01483
- 2017b, *PASP*, 129, 058004. 1701.08807
- Lacy, M., & Halstead, D. 2015, *ALMA data rates and archiving at the NAASC*, Tech. rep., NRAO. URL http://library.nrao.edu/public/memos/naasc/NAASC_110.pdf
- Madura, T. I. 2017, *PASP*, 129, 058011. 1611.09994

- Myers, S. T., Law, C., Chandler, C., & Lacy, M. 2015, VLA Technical Working Group. URL https://safe.nrao.edu/wiki/pub/JVLA/TechnicalWorkingGroup/Technical_Implementation_Plan_PDR.pdf
- Naiman, J. P. 2016, *Astronomy and Computing*, 15, 50. 1602.03178
- Naiman, J. P., Borkiewicz, K., & Christensen, A. J. 2017, *PASP*, 129, 058008. 1701.01730
- Punzo, D., van der Hulst, J. M., Roerdink, J. B. T. M., Oosterloo, T. A., Ramatsoku, M., & Verheijen, M. A. W. 2015, *Astronomy and Computing*, 12, 86. 1505.06976
- Springel, V. 2005, *MNRAS*, 364, 1105. astro-ph/0505010
- Taylor, R. 2015, *Astronomy and Computing*, 13, 67. 1510.03589
- 2017, *PASP*, 129, 028002. 1611.02517
- Terlouw, J. P., & Vogelaar, M. G. R. 2016, Kapteyn Package: Tools for developing astronomical applications, *Astrophysics Source Code Library*. 1611.010
- Tully, R. B., Rizzi, L., Shaya, E. J., Courtois, H. M., Makarov, D. I., & Jacobs, B. A. 2009, *AJ*, 138, 323
- Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., & Norman, M. L. 2011, *ApJS*, 192, 9. 1011.3514
- Vogt, F. P. A., Seitzzahl, I. R., Dopita, M. A., & Ruiter, A. J. 2017, *PASP*, 129, 058012. 1611.03862
- Vogt, F. P. A., & Shingles, L. J. 2013, *Ap&SS*, 347, 47. 1305.5534
- Vohl, D., Fluke, C. J., Hassan, A. H., & Barnes, D. G. 2017a, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of *Astronomical Society of the Pacific Conference Series*, 507. 1610.00806
- Vohl, D., Fluke, C. J., Hassan, A. H., Barnes, D. G., & Kilborn, V. A. 2017b, in *Astroinformatics*, edited by M. Brescia, S. G. Djorgovski, E. D. Feigelson, G. Longo, & S. Cavuoti, vol. 325 of *IAU Symposium*, 311. 1612.00920



Brian Kent answering questions, with session (and POC) chair Nuria Lorente. (Photo: Keith Shortridge)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

An Introduction to FITSWebQL

C. Zapart,¹ Y. Shirasaki,¹ M. Ohishi,¹ Y. Mizumoto,¹ W. Kawasaki,¹
 T. Kobayashi,¹ G. Kosugi,¹ E. Morita,¹ A. Yoshino,¹ and S. Eguchi²

¹*National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, Japan; chris.zapart@nao.ac.jp*

²*Fukuoka University, 8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan*

Abstract. The JVO ALMA WebQL web service - available through the JVO ALMA FITS archive - has been upgraded to include legacy data from other telescopes, for example Nobeyama NRO45M in Japan. The updated server software has been renamed FITSWebQL. In addition, a standalone desktop version supporting Linux, macOS and Windows 10 Linux Subsystem (Bash on Windows) is also available for download from <http://jvo.nao.ac.jp/~chris/>.

The FITSWebQL server enables viewing of even 100GB-large FITS files in a web browser running on a PC with a limited amount of RAM. Users can interactively zoom-in to selected areas of interest with the corresponding frequency spectrum being calculated on the server in near real-time. The client (a browser) is a JavaScript application built on WebSockets, HTML5, WebGL and SVG.

There are many challenges when providing a web browser-based real-time FITS data cube preview service over high-latency low-bandwidth network connections. The upgraded version tries to overcome the latency issue by predicting user mouse movements with a Kalman Filter in order to speculatively deliver the real-time spectrum data at a point where the user is likely to be looking at. The new version also allows one to view multiple FITS files simultaneously in an RGB composite mode (NRO45M FUGIN only), where each dataset is assigned one RGB channel to form a color image. Spectra from multiple FITS cubes are shown together too.

The paper briefly describes main features of FITSWebQL. We also touch on some of the recent developments, such as an experimental switch from C/C++ to Rust (see <https://www.rust-lang.org/>) for improved stability, better memory management and fearless concurrency, or attempts to display FITS data cubes in the form of interactive on-demand video streams in a web browser.

1. Introduction

Historically the ALMA WebQL service offered by the Japanese Virtual Observatory dates back at least to the year 2012 when it was presented during the ADASS XXII Conference (Eguchi et al. 2013). Afterwards, in order to keep up with ever growing FITS file sizes coming out of the ALMA observatory and also to offer improved functionality, newer versions have been released on a regular basis. For example, released in 2017, version 3 introduced an experimental 3D view of FITS data cubes. In 2018 the current version 4 — completely re-written from scratch in the Rust programming language — features real-time streaming videos of individual frequency chan-

nels in the FITS data cubes. The service can be accessed from the JVO Portal, found at <https://jvo.nao.ac.jp/portal/top-page.do>. The latest version 4 of the software (which includes the standalone desktop edition) is freely available from the following GitHub repository: https://github.com/jvo203/fits_web_q1. The unchanging motivation behind this web service is to provide a FITS file preview (quick look) and cut-out capability through a web browser. The service allows end users to view over 100GB-large FITS files in a web browser without ever having to download the underlying FITS files. After previewing FITS files users may choose to download interesting FITS files either in whole or to stream a partial region-of-interest (cut-out) from the JVO server to their own computers.

2. Architecture

The new version 4 initially started as a small feasibility study to find how easy it would be to re-implement the server part of FITSWebQLv3 in Rust. There are good reasons for switching from C/C++ to a new systems programming language such as Rust as it brings important benefits such as memory safety (*no memory leaks*), thread safety (*no data races*), better (smoother) multithreading compared with OpenMP in C/C++ and a complete lack of segmentation faults (*no crashes*) due to inherent safety measures built into the Rust language. It is certainly possible to write C/C++ programs that are free of memory leaks and do not crash but from a programmer's standpoint Rust makes accomplishing these tasks much easier, all without sacrificing performance. In addition, Rust has an integrated HTTP/WebSockets networking library: *actix-web* that compares favorably with the previously used disparate mix of C *libmicrohttpd* and C++ *µWebSockets*.

With the Rust port under-way work had also been progressing on adding streaming video capability to the main v3 C/C++ codebase. However, once all the bottlenecks with Rust have been identified and dealt with, another benefit has come to light: the original C/C++ codebase has become rather complex and adding new functionality has turned into an error-prone process running the risk of introducing memory leaks and bugs. Hence a decision has been taken to complete the switch from C/C++ to Rust and add streaming video functionality to the new version v4, of which the full client-server architecture can be seen in Figures 1- 2.

A two-way communication between the client (a web browser) and the Rust server occurs over WebSockets, which halve the network latency and are more efficient in handling small messages compared to traditional AJAX HTTP requests. On the server side, the Rust language binds together various C/C++ libraries for which there is no high-performance 100% pure-Rust implementation available. In particular, the computation-intensive parts are SIMD-parallelized using the Intel SPMD Program Compiler.¹ Unfortunately the *no-crash* guarantees do not extend to external non-Rust libraries which may leak memory and may contain segmentation fault-causing bugs. One needs to be very careful when choosing which C/C++ libraries to call from Rust.

On the client side we have taken advantage of the latest developments in browser technologies: the widespread adoption of WebAssembly (Wasm) that allows developers

¹The open-source Intel SPMD compiler (see <https://ispc.github.io>) should not be confused with the paid-for Intel C/C++/Fortran compiler suite.

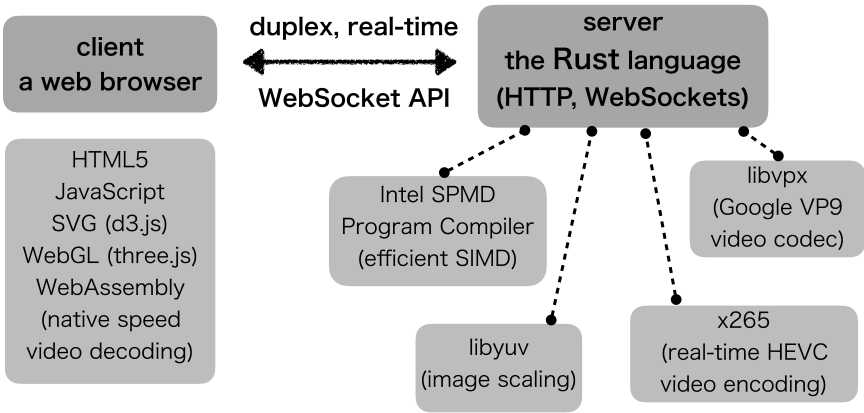


Figure 1. FITSWebQLv4 client-server architecture.

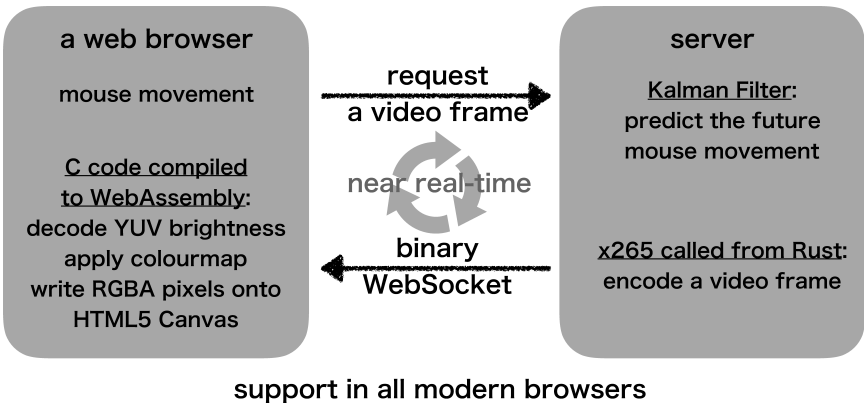


Figure 2. WebAssembly (Wasm) acceleration (near-native speed execution).

to compile C/C++ code to a binary Wasm stack machine code executed at a near-native speed inside a web browser.² In particular the CPU-intensive parts: real-time decoding of HEVC video frames and the application of a user-specified colormap to greyscale video frames have been greatly accelerated with WebAssembly.

2.1. VP9 vs. HEVC comparison

During the initial development originally the author intended to use the HEVC (via its x265 encoder library) codec to handle real-time video streams. However, finding a suitable JavaScript and/or Wasm decoder has proved impossible. The resources freely available on the Internet did not meet our requirements. They were too outdated; they did not support the latest HEVC specification. As an alternative, after exploring other

²<https://webassembly.org>

codecs i.e. Cisco’s Thor, initially we integrated Google’s VP9 libvpx library into our project. However, due to inferior multithreading capabilities of libvpx and codec inefficiencies compared with a superior HEVC solution, we decided to return to using HEVC/x265. Since there was no suitable off-the-shelf HEVC browser decoder available, we were forced to adapt the HEVC decoding part from the FFmpeg C library and compile it to WebAssembly for fast native execution in a web browser.

As a result of this somewhat convoluted development process, as of now the VP9 library is still used to compress FITS 2D images (as VP9 still keyframes) for display in a browser whilst the more capable HEVC x265 library handles real-time video streaming. Table 1 shows the main pros and cons of the two codec formats.

Table 1. A side-by-side comparison of Google’s VP9 and HEVC video codecs together with their corresponding C API libraries.

Google’s VP9 (libvpx)	HEVC (x265)
libvpx library: both an encoder and decoder	x265 library: only an encoder (search the Internet for a decoder to suit your task)
slower, less efficient encoding, inferior multithreading	faster than libvpx, more efficient (bandwidth-friendly), scales well across all CPU cores
no greyscale (an overhead of handling redundant RGB/YUV channels)	YUV 4:0:0 support (server-encode as greyscale, add color in the client)
an easy API, trivial to compile the decoder into WebAssembly	extreme difficulty finding a suitable JavaScript decoder (DIY: FFmpeg C API compiled to WebAssembly)

3. Conclusions

Based on our experience at JVO Rust has largely lived up to its promises. Not only did performance not deteriorate, in a few places it has actually improved compared to the C/C++. The only disadvantage of Rust seems to be its steep learning curve.

References

Eguchi, S., Kawasaki, W., Shirasaki, Y., Komiya, Y., Kosugi, G., Ohishi, M., & Mizumoto, Y. 2013, in *Astronomical Data Analysis Software and Systems XXII*, edited by D. N. Friedel, vol. 475 of *Astronomical Society of the Pacific Conference Series*, 255. 1211. 3790

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

An HDF5 Schema for SKA Scale Image Cube Visualization

A. Comrie,^{1,2} A. Pińska,^{1,2} R. Simmonds,^{1,2} and A. R. Taylor^{1,2,3}

¹*Inter-University Institute for Data Intensive Radio Astronomy, South Africa;*
angus@idia.ac.za

²*University of Cape Town, Cape Town, South Africa*

³*University of the Western Cape, Cape Town, South Africa*

Abstract. We describe work that has been performed to create an HDF5 schema to support the efficient visualization of image cubes that will result from SKA Phase 1 and precursor observations. The schema has been developed in parallel to a prototype client-server visualization system, intended to serve as a testbed for ideas that will be implemented in systems developed to replace the existing CyberSKA and CASA viewers.

1. Introduction

Most astronomy image files are currently packaged using the FITS standard (Wells & Greisen (1979)); however, the FITS standard is not well-suited for storing or defining additional derived data products in a hierarchical structure. The HDF5 technology suite (Folk et al. (2011)) provides a data model, file format, API, library, and tools to enable the creation of structured schemas for different applications. We will show how these can be beneficial in packaging of radio astronomy (RA) data. In particular, our interest is in supporting fast interactive visualization of data that will be produced by the SKA telescope and its precursors.

Existing HDF5 schemas developed for RA data did not meet our requirements. The LOFAR HDF5 schema (Anderson et al. (2010)) did not meet performance requirements, because of its approach of storing each 2D image plane in a separate group. The HDFITS schema (Price et al. (2015)) serves as a starting point for an HDF5 schema that maintains round-trip compatibility with the FITS format, but lacks the additional structures required for pre-calculated and cached datasets. We have therefore created a new schema tailored to our application, although it may be advantageous for other processing and analysis applications as well. The schema is similar to that of HDFITS, but extensions have been added to support a number of features required for efficient visualization of large data sets.

2. Requirements

For our application, we use client-server visualization tools to view large RA image cubes remotely. We are currently working with data from MeerKAT Large Science

Projects, but aim to have a tool that will scale to the the data produced by the SKA. This scale of data needs to be maintained on compute clusters with large, fast storage systems: it is too large to download to workstations.

The server-based tool needs to be able to load image cubes quickly and to be able to start manipulating them without a large amount of initial processing. Reproducible results of commonly used compute or I/O intensive tasks therefore need to be pre-calculated and stored in a hierarchical structure for easy access. To support these aims, a number of different types of datasets are required:

- **Average datasets** store the average of the dataset along a particular axis (normally Z). These generally have a higher signal-to-noise ratio, and are useful during data visualization. Calculating them on the fly is computationally expensive. The name of the axis along which the average is taken should be indicated by the dataset name.
- **Permuted datasets** (e.g. $XYZ \rightarrow ZYX$) will allow for enormous speedups when reading image slices along non-principal axes. The schema defines how optional permuted datasets are stored in a standardized manner, so that software supporting the schema can check for these datasets when performing I/O-intensive dataset slices, such as reading a Z -profile at a given (X,Y) pixel value. The name of the permuted dataset should indicate the permuted layout.
- **Mip-mapped datasets** store a copy of the dataset, down-sampled across a particular image plane (e.g XY). As the visualization of large data generally requires down-sampling of generated images to match the user's viewport, this allows for the visualization of large data sets without loading entire image planes and performing down-sampling for each generated image. In addition, this will enable an efficient delivery of images to the client using tiling techniques commonly used in geographic information system (GIS) applications.
- **Histograms** defined along a particular image plane (e.g. XY or YZ) are I/O intensive to calculate, but relatively small and simple to store. For example, calculating the histogram for a 4096×4096 image slice takes approximately 80 ms of calculation time, while calculating the histogram for an entire cube can take far longer. Using the "square root" guideline, a histogram with $N = 4096$ would take an additional 16 KB of storage space. Stored histograms can be used to calculate approximate percentile values, which are commonly used in image visualization to restrict color mapping to a range of the data values, thus preventing outliers from skewing the color-mapped image. Approximate percentiles are sufficient for our purposes, provided that the number of histogram bins is large enough.

3. Schema

Initial tests of the schema are based on files converted from existing FITS files produced by scientists working on the MeerKAT data pipeline. When a FITS file is converted, a top-level group called \emptyset , which corresponds to the first Header Data Unit (HDU) in the original file, is created. Additional HDUs are stored in sequentially numbered groups, with the name of the HDU stored as the **NAME** attribute of each top-level group. The FITS data is saved as the **DATA** dataset of each top-level group.

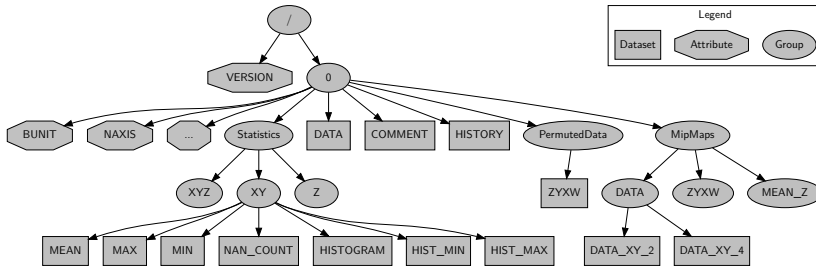


Figure 1. Outline of our HDF5 schema, indicating the structure of basic attributes, datasets and additional precomputed data.

FITS header entries for each HDU are stored as attributes of the relevant top-level group. This allows files in the schema to be translated back into FITS format if needed. We translate the `COMMENT` and `HISTORY` attributes to datasets rather than multidimensional attributes.

We show an outline of our proposed schema for storing these additional features in the HDF5 file in Figure 1. The name of each permuted dataset indicates the permuted order of axes. In the case of the example shown, a 4D cube (XYZ with the W -coordinate being Stokes parameters) has an additional dataset stored with the Z - and X -coordinates permuted. Statistics and mipmapped datasets are named as shown, with the mip factor indicated by the suffix of the dataset name.

A file will generally contain only a selection of the above additional features, depending on the application. We can strip features out by copying datasets selectively when offering downloads to clients. For example, permuted datasets and mipmaps are stored purely for performance reasons, and can be removed when we offer a download to clients, to minimize file size.

4. Performance

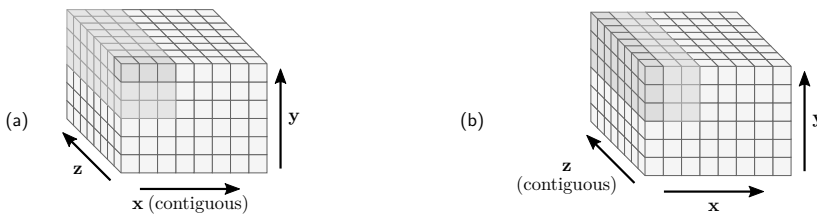


Figure 2. Example of accessing a $3 \times 3 \times 8$ region of an image cube with dimensions $8 \times 6 \times 8$, with the contiguous coordinate chosen as (a) X and (b) Z .

In the example shown in Figure 2, a spectral profile is calculated from a $3 \times 3 \times 8$ region of an image cube with dimensions $8 \times 6 \times 8$. In the standard approach, when the X -coordinate is contiguous, each read operation consists of a $3 \times 4 = 12$ byte read, followed by a seek to the next row, yielding a total of $3 \times 8 = 24$ read operations. When we use the permuted dataset, with the Z -coordinate contiguous, each read operation

Table 1. Performance comparison of workloads on a $5850 \times 1074 \times 376$ image.

Workload	Original [ms]	Permuted [ms]	Speedup
YZ-slice	5033 ± 7	2.08 ± 0.05	2420 ± 50
Z-profile	52.9 ± 0.1	0.182 ± 0.002	291 ± 4
$32 \times 32 \times 376$ region	340.1 ± 1.0	5.27 ± 0.04	64.5 ± 0.5
$64 \times 64 \times 376$ region	604.4 ± 1.7	13.1 ± 0.1	46.0 ± 0.5
$128 \times 128 \times 376$ region	876.8 ± 2.9	54.4 ± 0.7	16.1 ± 0.2

now consists of a $3 \times 8 \times 4 = 96$ byte read, followed by a seek to the next column, yielding a total of 3 read operations. For small read sizes, disk throughput is bounded by the total number of I/O operations per second. Therefore, reducing the number of read operations will dramatically increase disk throughput.

Several features inherent to HDF5 provide performance advantages. The Parallel HDF5 library provides support for multiple processes to write to a single HDF5 file simultaneously using MPI. This can speed up image cube generation. In addition, HDF5 allows us to alter dataset chunk size without changing the schema or code required to read the data, as this is abstracted by the HDF5 interface. This means that we can performance-tune different files to suit different common access patterns while maintaining schema compatibility.

Table 1 compares the execution time of common imaging workloads when data is read from the original dataset and from a permuted copy. All measurements were performed using an NVMe SSD, and the system buffer cache was cleared between each benchmark run to ensure that results were not skewed by operating system-controlled caching. Significant speedups are seen in all tested workloads when a permuted dataset is used, with the readout of YZ plane-aligned subsets being most affected. The speedup reduces as the size in the X and Y dimensions increases, which indicates that for regions above a threshold the original dataset should be utilized for maximum efficiency.

5. Summary

In this paper we have presented a new HDF5 schema for astronomical image data. We have explained our motivation for creating this schema to support our requirements for the visualization of large data from radio astronomy. We have provided an overview of the schema and the types of data access patterns that it supports. Initial tests of reading from a permuted dataset defined in the schema show significant benefits for commonly performed workloads.

References

Anderson, K., Alexov, A., Baehren, L., Griessmeier, J.-M., Wise, M., & Renting, A. 2010, PoS, ISKAF2010, 062. [ASP Conf. Ser.442,53(2011)], 1012.2266
Folk, M., Heber, G., Koziol, Q., Pourmal, E., & Robinson, D. 2011, in Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases (ACM), 36
Price, D., Barsdell, B., & Greenhill, L. 2015, Astronomy and Computing, 12, 212
Wells, D. C., & Greisen, E. W. 1979, in Image Processing in Astronomy, 445

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Analysis of Astronomical Data using VR: the Gaia Catalog in 3D

Emanuel Ramírez,¹ Juan González Núñez,² José Hernandez,³ Jesús Salgado,⁴ Alcione Mora,⁵ Uwe Lammers,³ Bruno Merín,³ Deborah Baines,⁴ Guido de Marchi,⁶ and Christophe Arviset³

¹*Quasar Science Resources S. L., Edificio Ceudas, Ctra. de La Coruna Km 22.300, 28232, Las Rozas de Madrid, Madrid, Spain eramirez@quasarsr.com*

²*SERCO for ESA*

³*ESA, ESAC*

⁴*Quasar Science Resources for ESA*

⁵*AURORA for ESA*

⁶*ESA, ESTEC*

Abstract. Since 2016, the ESAC Science Data Center have been working on a number of Virtual Reality projects to visualize Gaia data in 3D. The Gaia mission is providing unprecedented astrometric measurements of more than 1 billion stars. Using these measurements, we can estimate the distance to these stars and therefore project their 3D positions in the Galaxy. A new application to analyze Gaia DR2 data is currently in development and planned to be publicly released soon. In this presentation we will give a demo of the latest version of the Oculus Rift application and will show specific use cases to analyze Gaia DR2 data as well as a demonstration on how Virtual Reality can be integrated into a data analysis workflow. We will also show how can new input techniques such as hand-tracking can bring new levels of freedom in how we interact with data.

1. Introduction

We aim to provide an innovative science-driven tool to explore astronomical data using Virtual Reality headsets and Hand-Tracking devices for interaction. Virtual Reality brings an unprecedented level of immersion in a 3D virtual space which gives the possibility to better explore multi-dimensional data and build a virtual work environments that are more flexible and intuitive to use.

While the present common data analysis tools used in Astronomy rely on 2D visualizations for the representation of data plots in 3D, native 3D visualizations that take advantage of the depth perception can lead to more precise selections, better clustering determinations, and overall quicker and more efficient interaction through tactile and haptic controllers. Also interaction with time, as when propagating proper motions and radial velocities for source catalogs, can be better perceived by the astronomer having depth perception of the visualizations involved.

2. Visualization

Unity3D (Multi-platform game engine) was selected as the development platform due to its agile development cycles and its integration with the Oculus SDK and Leap Motion SDK.

2.1. Displaying a star Catalog

Different techniques to build a 3D model of a star catalog were tested and the most effective solution found was the creation of meshes. On these meshes, each of the vertices represents the source's position in 3D space and are displayed using a small 2D texture graphic as a star.

By adjusting the distance between the two rendering cameras in Unity, the depth perception can be altered for an improved feeling of volume in Virtual Reality

2.2. Labels and PM vector lines

Using Unity’s UI system we place sprites next to the desired stars to display information relative to them, this can be any parameter included in the catalog or table. Using the same system we can also plot their PM vector lines on a 2D plane with its normal aligned with the origin of coordinates of each catalog.

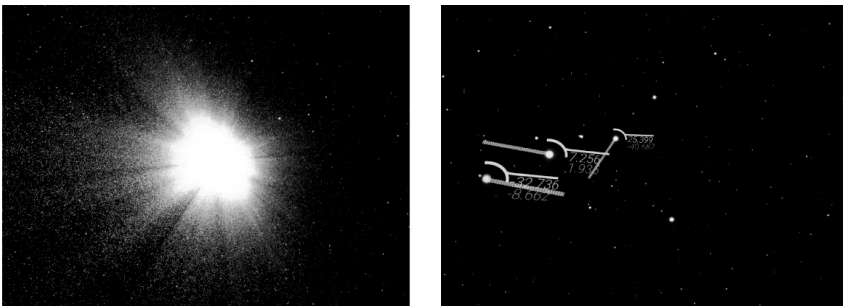


Figure 1. *Left:* Full TGAS Catalog as a several 3D meshes *Right:* Labels and PM vector lines for sources

2.3. Attributes Display

Source's parameters can be showed by using visual attributes like size, color and brightness of each star. These can be adjusted to represent real brightness, dimensions or color parameters, or as any combination for data visualization.

3. Interaction

Using Leap Motion hand tracking sensors we can map hand gestures and actions to interact with the data. To display the loaded catalogs we simply look at our left palm. We can grab any of the cubes (DataBlocks) by using our right hand and place them anywhere on our virtual space. Once the cube is released it displays its plotting options and stays in its current location until its grabbed again.

3.1. DataBlocks

Each of the loaded catalogs creates a "DataBlock", a small cube that can be picked up and moved to the desired visualization location. Once it's dropped, the visualization opens up and allows interaction. As well as activating an interaction panel for the plot.

Each Data Block can produce Spherical Plots and Cartesian Plots. We are currently developing additional plots such as 2D plots and Sky plots.

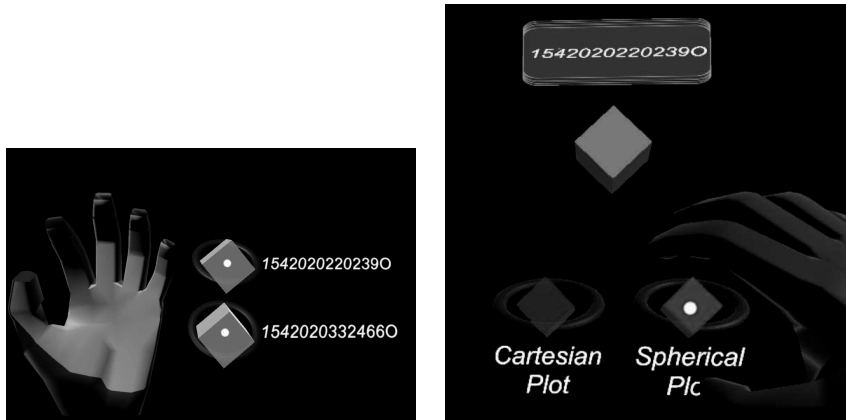


Figure 2. *Left:* DataBlock created from Queries to the Gaia Archive *Right:* Open DataBlock menu, showing two plotting options.

3.2. Control Panel

To enable more complex interactions with the data, each plot comes with a control panel with context-specific buttons.

Buttons using hand tracking are not as straight forward as in 2D interfaces, they need to have an appropriate response to proximity and activation, this is displayed by making them change colors depending on their current state. When a finger is nearby the border changes color to notify the possibility to press it.

Both plots allow for axis selection using the left and right buttons as well as Proper Motion lines using the PM button. The control panel is in continuous development and in future versions it will allow a lot more interaction.

3.3. Scaling and Rotating

With the use of one hand we can rotate the plot just by grabbing and moving like we would a real physical object. The plot rotates around its fixed center point.

By grabbing the plot with two hands we can scale the object by separating or joining them. This gesture seemed as the most intuitive, giving the feeling of expanding a physical object.

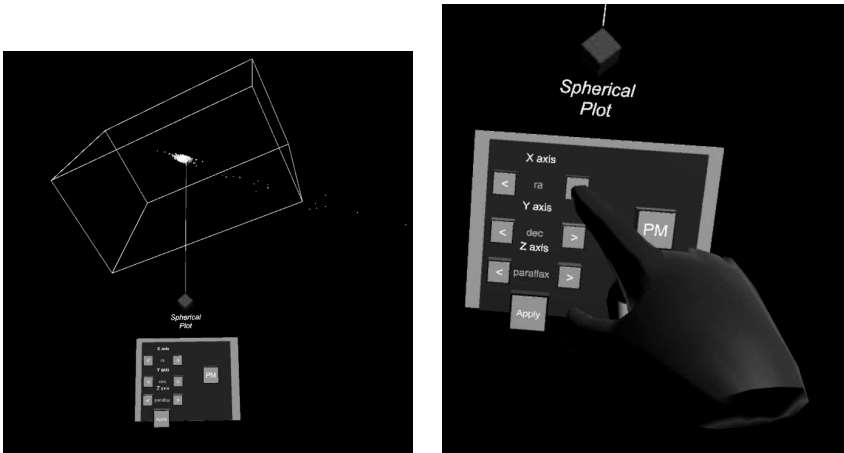


Figure 3. *Left:* Spherical Plot from a Gaia Archive Query *Right:* Control Panel for that Spherical Plot

4. Interoperability with other Astronomical Tools

Through the use of SAMP we can interact with other SAMP-enabled applications as well as the Gaia Archive and its web interface.

The application has the capability to run a SAMP Hub and a client to receive and send tables. Once a query has been made to the Archive and the application is running, we can create a DataBlock by pressing the "Send through SAMP" button on the Archive's Job. The DataBlock is added to the list on our left hand and now can generate plots and be interacted with.

5. Conclusions and Future Work

By mixing LeapMotion's hand tracking interaction with Virtual Reality we believe we can improve the data analysis workflow of astronomical data and bring new interactions that would be very complicated or impossible to produce in a conventional screen.

This application is in continuous development inside the European Space Data Center, over the last year it has had various changes taking into account incoming feedback from inside the ESDC, from other institutes and general users.

Among some of the development lines for upcoming features we're currently working on custom region selection techniques using hand gestures, animating the evolution of sources in a 3D space, better DataBlock management, better contextual information for the displayed data and many others.

We plan to release a beta version during the coming months with all the functionalities showed at ADASS.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Exoplanets Data Visualization in Multi-dimensional Plots using Virtual Reality in DACE

F. Alesina, F. Cabot, N. Buchschacher, and J. Burnier

*Observatoire astronomique de l'Université de Genève, 51 ch. des Maillettes,
1290 Versoix, Switzerland*

Abstract. The Data and Analysis Center for Exoplanets (DACE) is a web platform based at the University of Geneva (CH) dedicated to exoplanets data visualization, exchange and analysis.

This platform is based on web technologies using common programming languages like HTML and JavaScript. During the past 3 years, the plotting tools has been improved in order to display large datasets on the platform, dealing with browsers performances constraints.

The next challenge is to display the exoplanets data in multi-dimensional plots. The web virtual reality technology has been added on DACE, and allows the user to display the data in virtual reality devices like cardboards and headsets.

The virtual reality is used for displaying 3D plots of synthetic planetary populations, discovered exoplanets from different archives, and 3D planetary systems with a star and its orbiting planets.

The used technologies are webVR, external GPUs called eGPUs in order to increase laptop performances, HTC vive pro headset and google cardboards.

1. Introduction

The DACE platform provides advanced visualizations of multi-dimensional datasets working on the most popular web browsers. With the explosion of virtual reality applications, we decided to explore this technology in order to provides new data visualizations. This project was initiated by a master student in computer sciences, at the University of Geneva and is now fully integrated in DACE on 3 modules.

There are two kind of visualizations possible in virtual reality. The realistic ones, popular and used by people that are interested by the beautiful views provided by professional headsets, and the scientific ones, used by professionals in order to achieve a task more easily than with a screen.

All the following outputs are generated according to the parameters filled in the DACE website and the data available in the DACE database.

2. Planetary system visualization

It is possible to run a dynamical evolution simulation using the GENGA integrator on a Keplerian system found with radial velocities or light curves tools. A 3D animation can be generated with the integrator and displayed on a web page or on a Virtual Reality device, like Google cardboards or HTC Vive headset.

Since 2015, a WebGL animation is showing the short time evolution of the planetary system, with planet trajectories.

This animation was updated in 2018 in order to add a virtual reality output combined to WebGL. The result is a web page with a headset button on the bottom right corner. If a virtual reality headset is plugged on the computer, then the animation is exported to the device and a view is displayed on the computer.

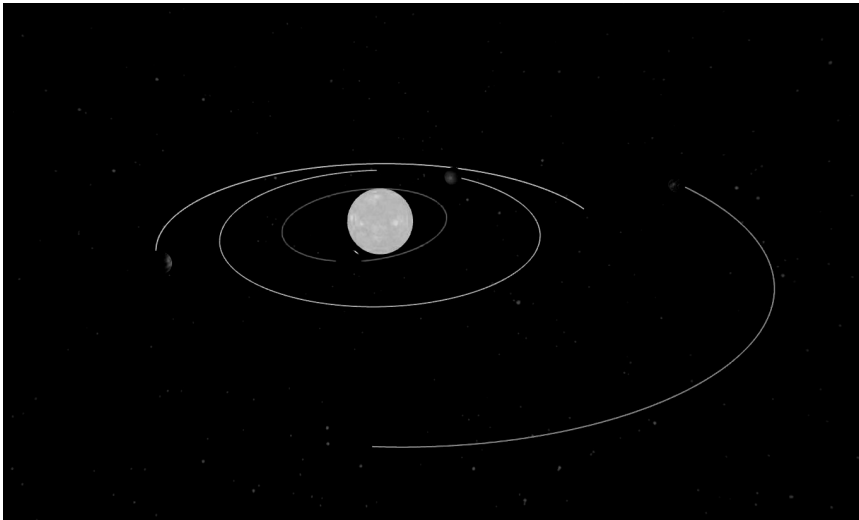


Figure 1. Screenshot of the GENGAs WebGL animation

The technologies used are WebGL, aFrame framework and HTC Vive Pro headset. The controllers of the HTC Vive Pro are used for changing the planetary system orientation.

3. Exoplanets table

After a successful technology test with the planetary visualization, the next step was to find how scientist could use virtual reality to do science. The Exoplanets Table module in DACE provides a table and plot view of this multi-dimensional data. It was chosen in order to test the implementation of a 3D-Virtual Reality library.

After few month of implementation and using the same technologies as the planetary system visualization, the 3D-Virtual Reality plot library was integrated in DACE. It provides a three dimensional view of the plot, and a lot of user interactions with the headset controllers like data selection, zooms and translations. Other display settings can be changed by the user like points sizes, axes displays, labels, guides etc.

4. Synthetic planetary populations

The Virtual Reality application was extended to the Synthetic Planetary Populations module. Synthetic populations generated by the University of Bern team are available

on DACE. We use these models to understand how planetary systems are created and to compare real observations and simulations.

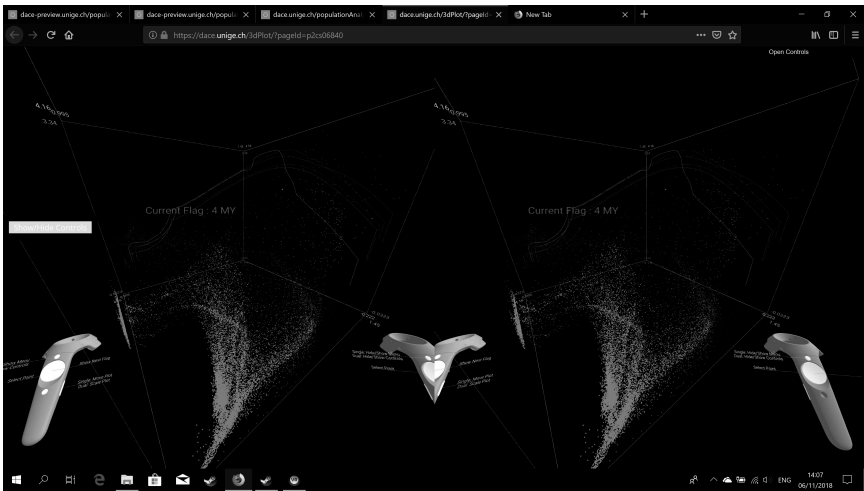


Figure 2. Screenshot of the synthetic planetary populations webGL visualization in virtual reality

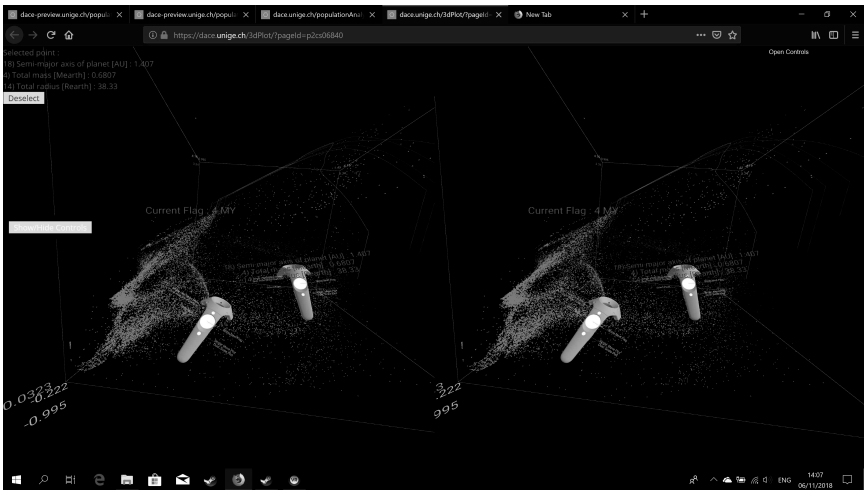


Figure 3. Screenshot of the synthetic planetary populations webGL visualization in virtual reality

With thousands of points, this plot is much bigger than the exoplanets table one. The synthetic populations are a set of snapshots taken at different ages. Each age is displayed one by one, using the buttons of the HTC Vive Pro controllers. It is also possible to display a track of a planet simulation by using the time as the fourth dimension.

Acknowledgments. This work has been carried out within the framework of the National Centre for Competence in Research PlanetS supported by the Swiss National Science Foundation. The authors acknowledge financial support from the SNSF. This publication makes use of DACE, a Data Analysis Center for Exoplanets, a platform of the Swiss National Centre of Competence in Research (NCCR) PlanetS, based at the University of Geneva (CH).



Felix Stoehr mesmerized in an exoplanet system (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

VisIVO Visual Analytics Tool: An EOSC Science Demonstrator for Data Discovery

Ugo Becciani,¹ Fabio Vitello,¹ Eva Sciacca,¹ Alessandro Costa,¹
 Antonio Calanducci,¹ Simone Riggi,¹ and Sergio Molinari²

¹*INAF, Catania Astrophysical Observatory, Catania, Italy;*
ugo.becciani@inaf.it

²*INAF - Institute for Astrophysics and Space Planetology, Rome, Italy*

Abstract. The Astrophysical community has set up a new suite of cutting-edge Milky Way surveys, spanning the electromagnetic spectrum, that provide a homogeneous coverage of the entire Galactic Plane and that have already started to transform the view of our Galaxy as a global star formation engine.

This paper presents the works devoted for the integration in the European Open Science Cloud (EOSC) of a visual analytics environment based on VisIVO (Visualization Interface for the Virtual Observatory). The application is investigating the use of the EOSC technologies for the archive services and intensive analysis employing the connection with a science cloud gateway.

1. Introduction

VisIVO¹ (Sciacca et al. 2015) is an integrated suite of tools and services for data discovery that include collaborative portals, mobile applications, visual analytics tool and a number of key components such as workflow applications, analysis and data mining functionalities. Space missions and ground-based facilities produce massive volumes of data and the ability to collect and store them is increasing at a higher pace than the ability to analyze them. This gap leads to new challenges in the analysis pipeline to discover information contained in the data.

VisIVO Visual analytics (Vitello et al. 2018) tool for star formation regions focuses on handling these massive and heterogeneous volumes of information accessing the data previously processed by data mining algorithms and advanced analysis techniques with highly interactive visual interfaces offering scientists the opportunity for in-depth understanding of massive, noisy, and high-dimensional data.

The aforementioned challenges demands an increasing archiving and computing resources as well as a federated and inter-operable virtual environment enabling collaboration and re-use of data and knowledge.

The European Open Science Cloud (EOSC) is constituting a large infrastructure to support and develop open science and open innovation in Europe and beyond. The EOSC is projected to become a reality by 2020 and will be Europe's virtual environment

¹VisIVO web page: <http://visivo.oact.inaf.it/>

for all researchers to store, manage, analyze and re-use data for research, innovation and educational purposes. The EOSCpilot project² is supporting the first phase in the development of the EOSC. It brings together stakeholders from research infrastructures and e-Infrastructure providers and engage with funders and policy makers to propose and trial EOSC's governance framework.

The VisIVO project has been selected as science demonstrator functioning as high-profile pilot that integrate astrophysical data and visual analytics services and infrastructures to show interoperability within other scientific domains such as earth sciences and life sciences. Thus, the connection with the European Open Science Cloud has been investigated exploiting the services developed within the European Grid Initiative (EGI) such as the ones to allow federated authentication and authorization and the federated cloud for analysis and archiving services.

2. VisIVO Visual Analytics

The ViaLactea Visual Analytic tool (VLVA) is an environment for the study of star forming regions on our Galaxy exploiting the connection with the ViaLactea Knowledge Base (VLKB) (Molinaro et al. 2016). It allows the access, analysis and visualization on a new set of complete and high spatial resolution Galactic Plane Surveys. Alongside the data collections it expose also the knowledge derived from the data including information related to e.g. filamentary structures, bubbles and compact sources.

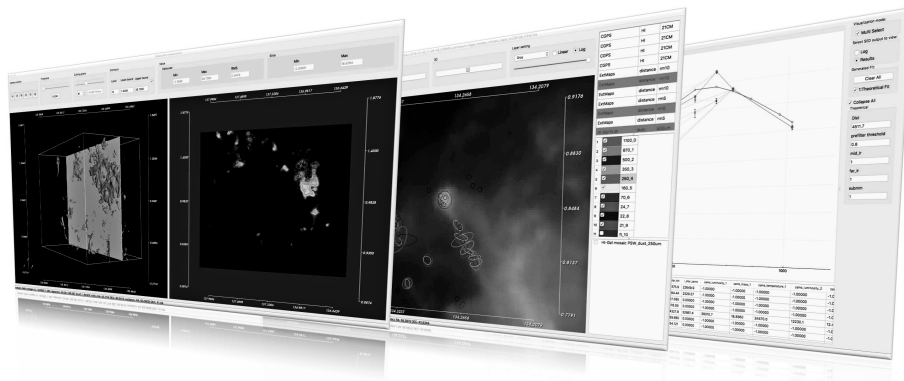


Figure 1. Some screen-shots of the VLVA tool: 3-D molecular spectral cubes (on the left), identification of compact sources (in the center), and, computation of spectral energy distributions (on the right).

The tool combines different types of visualization to perform the analysis exploring the correlation between different data, for example 2-D intensity images with 3-D molecular spectral cubes. Figure 1 shows some screen-shots of the tool. The scientist is enabled to discover the link between different physical structures, from the extended filamentary-shaped structures to the most compact, dense sources precursors of new stars. The implementation philosophy behind VLVA is to make transparent to

²EOSCpilot web page: <https://eoscspilot.eu/>

the scientist the access to all information without requiring technical skills to query the VLKB.

3. EOSC Science Demonstrator

The VLVA application has been extended by exploiting the use of the European Open Science Cloud technologies for the VLKB archive services and intensive analysis employing the connection with the ViaLactea Science Gateway³ (Sciaccia et al. 2017).

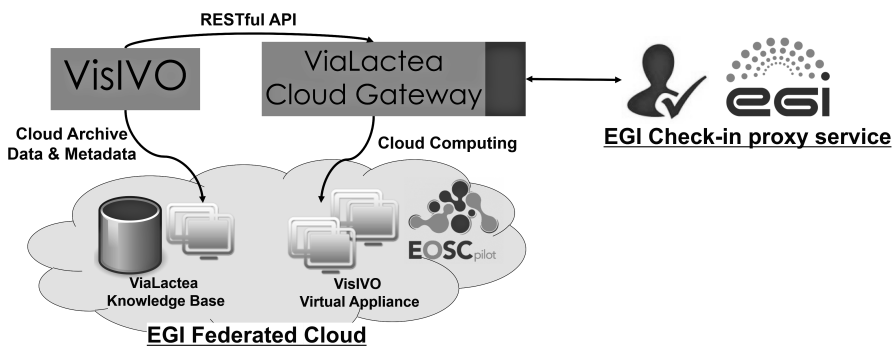


Figure 2. Architecture of the VisIVO EOSC Science Demonstrator implementation and employed services.

Figure 2 shows the overall architecture of the VisIVO EOSC Science Demonstrator implementation and employed services. The archiving services have been deployed within the EGI Federated Cloud toward the assurance of a FAIR access to the surveys data and related metadata. The science gateway has been integrated with the EGI Check-in⁴ service to enable the connection from the federated Identity Providers and with the EGI Federated Cloud⁵ to expand the computing capabilities making use of a dedicated virtual appliance stored into the EGI Applications Database⁶. Actually the virtual appliance is exploited for massive calculation of spectral energy distributions but may be expanded for other kind of analysis.

Furthermore, we have implemented also a lightweight version of science gateway framework developing an ad-hoc RESTful API to expose a simple set of functionalities to define pipelines and executing scientific workflows on any Cloud resources, hiding all the details of the underlying infrastructures.

³ViaLactea Science Gateway: <https://vialactea-sg.oact.inaf.it/>

⁴EGI Check-in service: <https://www.egi.eu/services/check-in/>

⁵EGI Federated Cloud: <https://www.egi.eu/services/cloud-compute/>

⁶EGI Applications Database: <https://appdb.egi.eu/store/vappliance/visivo.sd.va>

4. Conclusions and Future Works

We presented the effort done within the European Open Science Cloud initiative to extend visual analytics techniques and applications for the astrophysics community engaged in star formation studies. This project has produced the following outcomes: i) integration of visual analytics tools with EOSC services; ii) optimization of the archiving of multi-wavelength surveys under FAIR principles; iii) increase of computing resources for analysis (e.g. for calculation of spectral energy distributions); iv) a federated and interoperable virtual environment enabling collaboration and re-use of data and knowledge.

Recently, the development related to VisIVO has been exploited for the experimentation of cutting-edge interactive visualization technologies for the improvement of teaching and scientific dissemination. This work is being carried out as a knowledge transfer from astrophysical sciences to geological sciences in the context of an international collaboration to innovate teaching, learning and dissemination of earth sciences, using virtual reality (Gerloni et al. 2018).

Acknowledgments. The research leading to these results has received funding from the European Commissions Horizon 2020 research and innovation program under the grant agreement No. 739563 (EOSCPilot.eu).

References

- Gerloni, I. G., Carchiolo, V., Vitello, F. R., Sciacca, E., Becciani, U., Costa, A., Riggi, S., Bonali, F. L., Russo, E., Fallati, L., et al. 2018, in 2018 Federated Conference on Computer Science and Information Systems (FedCSIS) (IEEE), 527
- Molinaro, M., Butora, R., et al. 2016, in SPIE Astronomical Telescopes+ Instrumentation (International Society for Optics and Photonics), 99130H
- Sciacca, E., Becciani, U., Costa, A., Vitello, F., Massimino, P., Bandieramonte, M., Krokos, M., Riggi, S., Pistagna, C., & Taffoni, G. 2015, *Astronomy and Computing*, 11, 146
- Sciacca, E., Vitello, F., Becciani, U., Costa, A., Hajnal, A., Kacsuk, P., Farkas, Z., Marton, I., Molinari, S., Di Giorgio, A. M., et al. 2017, *Future Generation Computer Systems*
- Vitello, F., Sciacca, E., Becciani, U., Costa, A., Bandieramonte, M., Benedettini, M., Di Giorgio, A., Elia, D., Liu, S., Molinari, S., et al. 2018, *Publications of the Astronomical Society of the Pacific*, 130, 084503

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Open-source Web Tools for Spectroscopic and Imaging Data Visualization for the VOXAstro Initiative

Kirill A. Grishin,^{1,2} Igor V. Chilingarian,^{3,1} and Ivan Yu. Katkov^{4,1}

¹*Sternberg Astronomical Institute, M.V.Lomonosov Moscow State University, Universitetsky prospect 13, Moscow, 119234, Russia; kirillg6@gmail.com*

²*Faculty of physics, M.V.Lomonosov Moscow State University, 1 Vorobyovy Gory, Moscow, 119991, Russia*

³*Smithsonian Astrophysical Observatory, 60 Garden St., Cambridge, MA 02138, USA*

⁴*New York University Abu Dhabi, Saadiyat Marina District, Abu Dhabi, UAE*

Abstract. Here we present a set of flexible open-source tools for visualization of spectral and imaging data developed for the VOXAstro projects. Using open-source web visualization libraries we developed interactive viewers to display low- and high-resolution spectra of stars and galaxies, allowing one to view spectra having resolution up to $R=80000$ without putting a significant load on server and client sides, which is achieved by choosing the adaptive spectral binning window and dynamically pre-loading the datasets. We implemented a number of additional features like multiple spectra display, output of header info (e.g., stellar atmospheric parameters or stellar population properties of galaxies), display of emission lines decomposition parameters (fluxes, widths, etc.). The spectral viewers can be easily embedded into any archive or database web-site. We also present a cutout service that extracts data on the fly from the UKIDSS near-infrared imaging survey and generates colour composite RGB stamps, which we use, e.g., in the RCSED web-site as an embedded service. The service uses IVOA SIAP to access images, which it then cuts out on the fly using Astropy functions. In the coming years we plan to expand the capabilities of our spectroscopic and imaging visualization services and use them in future projects within VOXAstro.

1. Introduction

With the rapid development of astrophysical web services the question how to efficiently display datasets of various types has become crucial. We developed and deployed several tools for visualization of imaging and spectral datasets for the VOXAstro initiative. VOXAstro stands for Virtual Observatory tools for eXtragalactic Astrophysics. This initiative includes several projects such as the Reference Catalog of Spectral Energy Distribution (RCSED; <http://rcsed.sai.msu.ru/> Chilingarian et al. 2017), K-corrections calculator (<http://kcor.sai.msu.ru/> Chilingarian et al. 2010) and several stellar spectral libraries. Here we present a set of flexible open-source tools for visualization of spectral and imaging data. For all applications, we tried to reach the maximal flexibility and convenience in embedding them into *html* pages which makes possible re-using them in other projects.

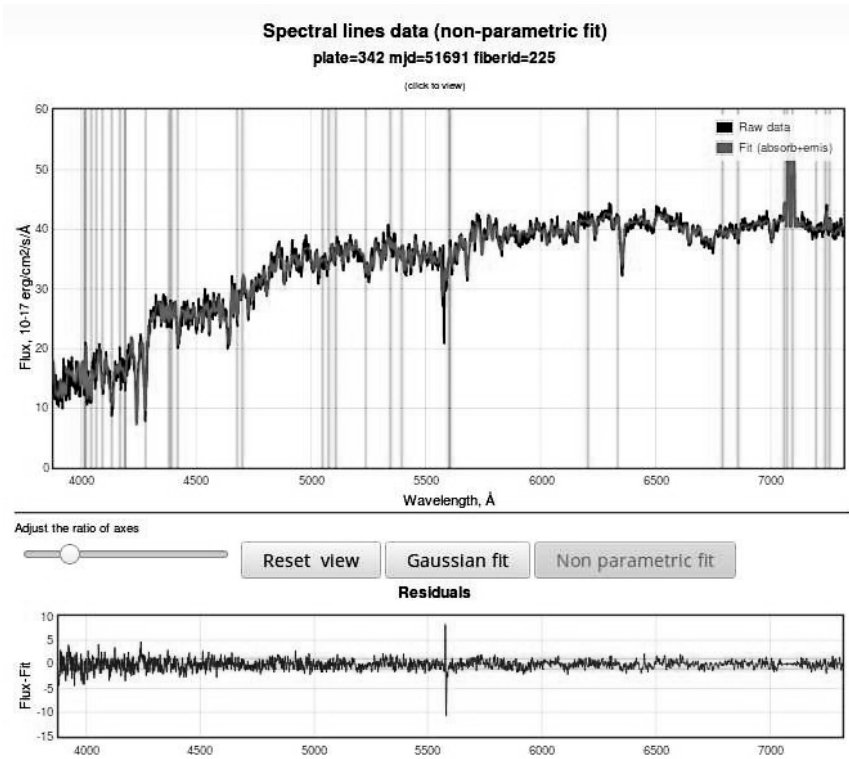


Figure 1. A galaxy spectrum from the RCSED project and its best-fitting template are displayed in the spectrum viewer. Fitting residuals are shown in the bottom panel.

An important aspect of data visualization in astrophysics arises from the fact that one needs to quickly generate plots, which potentially may contain tens of millions of data points. For example, it is often needed to plot a relatively small dataset of objects of interest over a 2D density plot for the entire reference sample of millions of sources, which always stay the same. This can be achieved using server side visualization to generate static plots, which are then displayed in a web-browser without the need of transmitting the entire dataset to the client. We successfully used this approach in the search of intermediate-mass black holes using RCSED (Chilingarian et al. 2018): for visual assessment of the complex multi-parametric selection we display several server-side generated plots reflecting various observed and derived properties of active galactic nuclei and their host galaxies – this resembles the well-known “connected views” paradigm of multi-dimensional data visualization.

2. A galaxy spectrum viewer in the RCSED project

For RCSED we developed a viewer of galaxy spectra. This application uses the `FLOR` JAVASCRIPT library (<https://www.flotcharts.org/>) which implements draggable

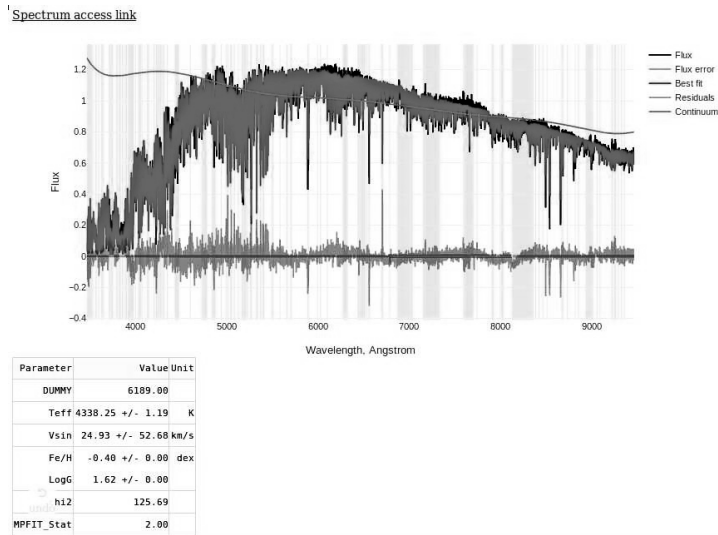


Figure 2. A spectrum of the star HD 163993 with its best-fitting model and residuals displayed in a simple spectrum viewer for the INDO-US stellar spectral library.

plots. The viewer (see Fig. 1) displays a calibrated spectrum and its best-fitting model (including models of emission lines and starlight). One can also get the information about fluxes in emission lines by clicking on them; the positions of lines are marked by vertical bars. Besides the spectrum plot, a clickable table is provided which contains emission line fluxes, flux errors, signal-to-noise ratios.

3. RGB stamp images for the UKIDSS near-infrared survey

Near-infrared RGB composite images are often useful for eyeball assessment of stellar populations and morphology in galaxies. The 2MASS survey (Skrutskie et al. 2006) gives access to all-sky JHK data, however it has relatively shallow depth because of short exposure times, therefore we use a much deeper UKIDSS (Lawrence et al. 2007) survey data in RCSED to study galaxies in detail. Unlike 2MASS, UKIDSS does not have an RGB MOC map in Aladin (Bonnarel et al. 2000), that is why it is non trivial to embed a UKIDSS RGB image to an arbitrary web page. We use the python library APLpy (<https://aplpy.github.io/>) to generate RGB images on the fly for the RCSED web-site. The Lupton et al. (2004) algorithm is used for the RGB composite generation from YHK data retrieved using IVOA Simple Image Access Protocol.

4. Interactive visualization for spectra of different types

In observational astrophysics one deals with spectral data of different flavors, from one-dimensional (like SDSS, Abazajian et al. 2009) to three-dimensional (e.g., MANGA, Bundy et al. 2015). The diversity of data one has to analyze makes it essential to develop a versatile spectral visualization tool. We developed a flexible interactive Web-

based tool for spectral display. It is based on Dash and Plotly libraries (<https://plot.ly/products/dash/>). These libraries allow us to efficiently display even high-resolution spectra ($R=80000$, 500k data points) without heavy load on either server or client. For end users it could also be important to display some values from a FITS file in a tabular form, e.g., stellar parameters. There is also an option to plot some arbitrary data as a supplementary dataset (e.g., a correcting polynomial for the stellar continuum). Supplementary datasets and the output parameter table are controlled via a URL. One can also adjust the parameters of the displayed datasets (flux, error, etc), which default to the IVOA Spectrum Data Model. This makes it possible to use our web application to visualization nearly all existing types of spectra.

5. Interactive visualization for libraries of stellar spectra

It is often needed to simultaneously visualize several spectra in order to compare them against each other. An important example is libraries of stellar spectra, which provide uniformly processed data collections from the same instrument. For this purpose, we extended our simple spectrum viewer. This tool inherits all its features but it also allows interactive selection of spectra from a data collection. For stellar libraries, we use the stellar atmospheric parameter space (T_{eff} , $\log g$, $[\text{Fe}/\text{H}]$). The stars can be selected from a 3D plot or from an interactive table, which contains stellar parameters.

Acknowledgments. This project is supported by the Russian Science Foundation Grant 17-72-20119. KG is grateful to the ADASS XXVIII organizing committee for providing financial aid to support his attendance of the conference.

References

- Abazajian, K. N., Adelman-McCarthy, J. K., Agüeros, M. A., Allam, S. S., Allende Prieto, C., An, D., Anderson, K. S. J., Anderson, S. F., Annis, J., Bahcall, N. A., & et al. 2009, *ApJS*, 182, 543-558. [0812.0649](#)
- Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. 2000, *A&AS*, 143, 33
- Bundy, K., Bershad, M. A., Law, D. R., Yan, R., Drory, N., & et. al., M. 2015, *ApJ*, 798, 7. [1412.1482](#)
- Chilingarian, I. V., Katkov, I. Y., Zolotukhin, I. Y., Grishin, K. A., Beletsky, Y., Boutsia, K., & Osip, D. J. 2018, *ApJ*, 863, 1. [1805.01467](#)
- Chilingarian, I. V., Melchior, A.-L., & Zolotukhin, I. Y. 2010, *MNRAS*, 405, 1409. [1002.2360](#)
- Chilingarian, I. V., Zolotukhin, I. Y., Katkov, I. Y., Melchior, A.-L., Rubtsov, E. V., & Grishin, K. A. 2017, *ApJS*, 228, 14. [1612.02047](#)
- Lawrence, A., Warren, S. J., Almaini, O., Edge, A. C., Hambly, N. C., Jameson, R. F., Lucas, P., Casali, M., Adamson, A., Dye, S., Emerson, J. P., Foucaud, S., Hewett, P., Hirst, P., Hodgkin, S. T., Irwin, M. J., Lodié, N., McMahon, R. G., Simpson, C., Smail, I., Mortlock, D., & Folger, M. 2007, *MNRAS*, 379, 1599. [astro-ph/0604426](#)
- Lupton, R., Blanton, M. R., Fekete, G., Hogg, D. W., O'Mullane, W., Szalay, A., & Wherry, N. 2004, *PASP*, 116, 133. [astro-ph/0312483](#)
- Skrutskie, M. F., Cutri, R. M., Stiening, R., Weinberg, M. D., Schneider, S., Carpenter, J. M., Beichman, C., Capps, R., Chester, T., Elias, J., Huchra, J., Liebert, J., Lonsdale, C., Monet, D. G., Price, S., Seitzer, P., Jarrett, T., Kirkpatrick, J. D., Gizis, J. E., Howard, E., Evans, T., Fowler, J., Fullmer, L., Hurt, R., Light, R., Kopan, E. L., Marsh, K. A., McCallon, H. L., Tam, R., Van Dyk, S., & Wheelock, S. 2006, *AJ*, 131, 1163

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Vissage: Viewing Polarization Data from ALMA

Wataru Kawasaki,¹ Yuji Shirasaki,¹ Christopher Andrew Zapart,¹
 Akira Yoshino,¹ Eisuke Morita,¹ Tsuyoshi Kobayashi,¹ George Kosugi,¹
 Masatoshi Ohishi,¹ and Yoshihiko Mizumoto¹

¹*National Astronomical Observatory of Japan, 2-21-1, Osawa, Mitaka, Tokyo, 181-8588, Japan; wataru.kawasaki@nao.ac.jp*

Abstract. Vissage¹ (*VISualisation Software for Astronomical Gigantic data cubEs*) is a Java-based standalone FITS browser (Kawasaki et al. 2013, 2014, 2017), primarily aiming to offer easy visualization of huge, multi-dimensional FITS data from ALMA. We report our recent implementation of its new capabilities of viewing polarization data and other minor updates.

1. Introduction

Some of the ALMA datasets available to the public in recent years contain polarization information other than Stokes I images. In addition to usual Stokes Q , U and V , images of linearly polarized intensity ($POLI = \sqrt{Q^2 + U^2}$) and polarization angle ($POLA = (1/2) \tan^{-1}(U/Q)$) are released as well in some datasets. These data are provided usually as separate FITS files, or packed in a single FITS file for some cases. Though these archive data should be useful for astronomers, however, software tools to easily view FITS-formatted polarization data, especially free ones, seem to be very rare or not exist.

For the very purpose of instantly viewing polarization images, we have been implementing new capabilities to Vissage, a new generation FITS browser under development, which compensates the limited functionality of the quick-look system FITSWebQL (Zapart et al. 2019) in JVO (Japanese Virtual Observatory, see Shirasaki et al. (2017)). Still in a primitive shape, we are trying to provide a simple and intuitive user interface so that even users not familiar with data cubes can easily access to polarization images.

2. Viewing Polarization Images

To display polarization images on Vissage, all you need to do is to drop FITS files and then select a value in a polarization menu:

- **Step 1** - Drag FITS files and drop them on Vissage

¹available from <http://jvo.nao.ac.jp/download/Vissage>

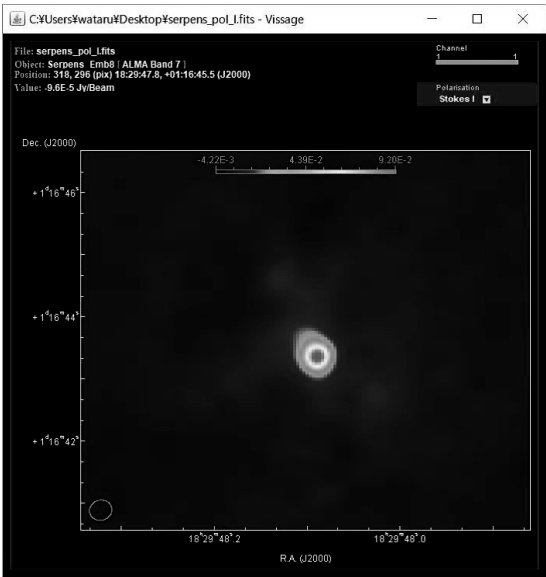


Figure 1. Showing polarization indicator above image.

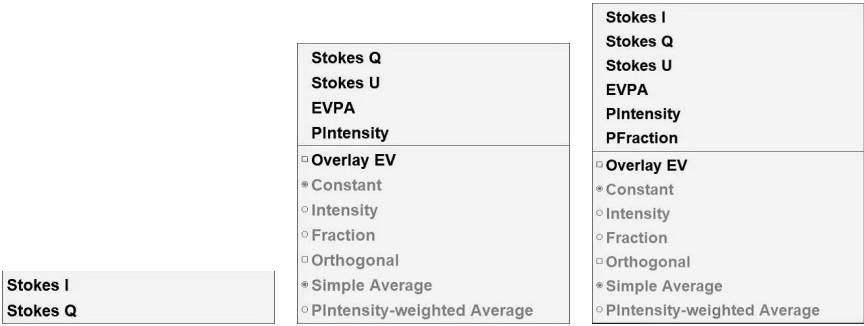


Figure 2. Polarization menus when Stokes *I* and *Q* images are dropped (Left), *Q* and *U* dropped (Middle) and *I*, *Q* and *U* dropped (Right), respectively.

If the dropped FITS files have a common coverage and resolution for both spatial and frequency axes and also a common observation date, they will be recognized as a single dataset, then one of the dropped data will be displayed. If the FITS file displayed first contains multiple Stokes data, the value stored at the first along the Stokes axis, probably Stokes *I* in most cases, should be displayed. If there are data with different coverage or resolution or observation date, they will be treated as a different dataset and displayed in a separate view. Once FITS files having Stokes axis is dropped, a polarization indicator appears showing which value is being displayed. The polarization indicator will be clickable if multiple polarization values are dropped. When clickable, the polarization indicator has a button-like symbol in the right side and acts like a drop-down list (see Figure 1).

• **Step 2** - Select a value in the polarization menu

A menu (hereafter ‘polarization menu’) appears if you click on the polarization indicator. It lists all values available from the dropped data (see Figure 2). Note that some values can be calculated via two ways: $\sqrt{Q^2 + U^2}/I$ or $(POLI)/I$ for degree of linear polarization, $(1/2) \tan^{-1}(U/Q)$ or $(POLA)$ for linear polarization angle, $\sqrt{Q^2 + U^2 + V^2}$ or $\sqrt{(POLI)^2 + V^2}$ for total polarized intensity, and so on. In case both calculations are available for a value, we adopt the simpler one which uses *POLI* or *POLA*. Select one, and you can see the image you selected (see Figure 3 left).

You can drop another FITS file on already displayed image to add polarization data – the item of the polarization menu will then be updated and you will have more choices.

Once polarization angle (EVPA) becomes available, you can also instantly overlay them as vectors on any type of polarization images. The vectors are shown for grid with size equal to beam area. Being a temporary shape, but some menus related to EVPA appears inside the polarization menu, under a separator line; you can select how to compute the lengths (constant, proportional to polarized intensity, proportional to degree of linear polarization) and the angles (simple mean or polarized intensity weighted mean) of vectors. To help users see possible magnetic field, displaying vectors rotated 90 degrees from the original direction is available as well. We plan to update the user interface for overlaying vectors to a more smart one so that more detailed options will be controllable.

The right panel of Figure 3 shows an example of displaying polarization data on Vissage, including an experimental capability of overlaying linear polarized intensity as dirty contours.

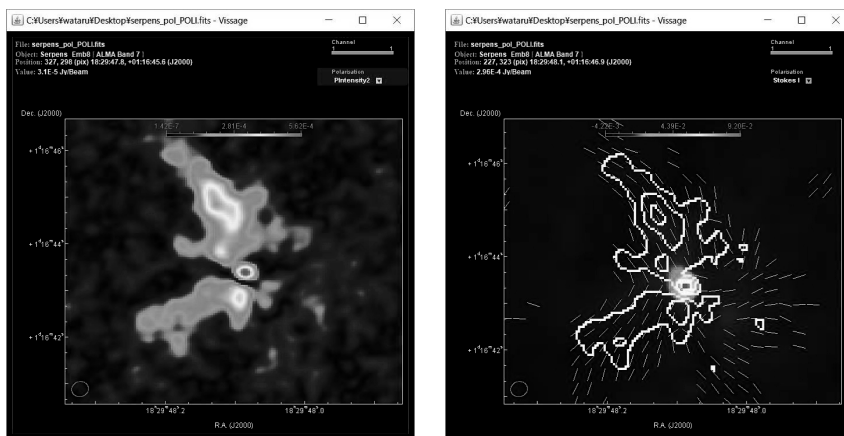


Figure 3. *Left*: Linear polarized intensity map. *Right*: Stokes I (color) + polarized intensity (contour) + polarization vector (90° rotated).

3. Other Minor Updates

Here are some of recent minor updates:

- Showing beam ellipse
- Showing colorbar
- Accessing to FITSWebQL versions 4 and 3 - available for FITS files downloaded from JVO only

4. Future Plans

The current capabilities of Vissage for polarization data are still primitive. Amongst many development items to expand and upgrade them, the followings are the ones with higher priority:

- Overlaying multiple arbitrary polarization images using colors and/or contours
- Rotation-Measure map
- Moment maps, channel maps, position-velocity diagram and so on, for polarization data
- Exporting polarization images to image files including Encapsulated PostScript (EPS) and other formats

References

- Kawasaki, W., Eguchi, S., Shirasaki, Y., Komiya, Y., Kosugi, G., Ohishi, M., & Mizumoto, Y. 2013, in ADASS XXII, edited by D. N. Friedel, vol. 475 of ASP Conf. Ser., 303
- 2014, in ADASS XXIII, edited by N. Manset, & P. Forshay, vol. 485 of ASP Conf. Ser., 285
- Kawasaki, W., Shirasaki, Y., Zapart, C. A., Kobayashi, T., Kosugi, G., Ohishi, M., Mizumoto, Y., Eguchi, S., Komiya, Y., & Kawaguchi, T. 2017, in ADASS XXV, edited by N. Lorente, K. Shortridge, & R. Wayth, vol. 512 of ASP Conf. Ser., 617
- Shirasaki, Y., Zapart, C. A., Ohishi, M., Mizumoto, Y., Kawasaki, W., Kobayashi, T., Kosugi, G., Kawaguchi, T., Eguchi, S., Ishihara, Y., Yamada, H., Hiyama, T., & Nakamoto, H. 2017, in ADASS XXV, edited by N. Lorente, K. Shortridge, & R. Wayth, vol. 512 of ASP Conf. Ser., 585
- Zapart, C. A., Shirasaki, Y., Ohishi, M., Mizumoto, Y., Kawasaki, W., Kobayashi, T., Kosugi, G., Morita, E., Yoshino, A., & Eguchi, S. 2019, in ADASS XXVIII, edited by P. Teuben, M. Pound, B. Thomas, & E. Warner, ASP Conf. Ser.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Realtime Telescope Data Visualization using Web Technologies

Pablo Mellado

Institute de RadioAstronomie Millimétrique, Granada, Granada, Spain;
mellado@iram.es

Abstract. When performing on-site or remote observations with a telescope, it is very critical to have a good feedback about the current status of your ongoing observation. Nowadays, web technologies have evolved to allow us to get data in real-time with the advantage of using just a web browser.

In our telescope we found the need of updating our previous outdated monitoring system which is currently showing information about the status of the telescope, the last scans plots, a view of the current weather conditions, etc.

This poster shows how the data visualization has improved by using newer technologies like microservices, websockets and messaging, as well as the structure developed to integrate them successfully in a reliable and more attractive way.

1. Introduction

When performing an observation with a telescope, it is essential for the user to know the current status of the telescope, especially the parameters related with the current observation. It is also very helpful to obtain previews of the results of every step that the user is carrying out.

Currently in our observatory all this information is provided by virtual desktops (VNC) which have several open windows with different information. This is not an ideal scenario because it is difficult to maintain and is time consuming.

Web technologies have evolved to a point where live data can be shown by just using a web browser. So we decided to create a web app that is easier to maintain and has a better layout of the information needed by the astronomer (Figure 1).

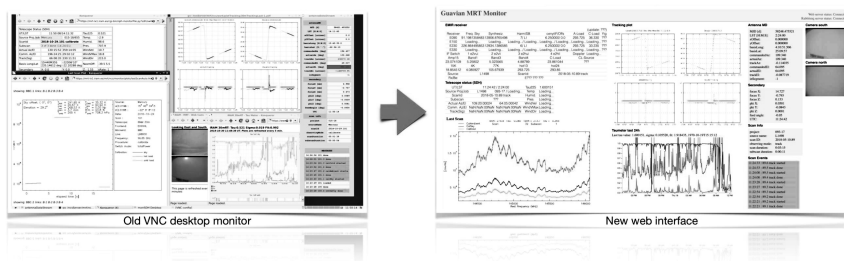


Figure 1. Old (left) and new (right) layouts for observatory web interface.

2. How it works

The new system gets the data following this process (see Figure 2):

- 1. The telescope control software continuously sends XML messages to the RabbitMQ server about its current status.
- 2. The messaging server saves some values to databases. It also forwards some of the messages to the web server according to the already specified subscription.
- 3. The web server is in charge of collecting all the relevant data for the astronomer. The telescope status is obtained from the messages but some plots and images are provided by microservices.
- 4. The images, plots and values are sent to the astronomer browser using websockets. This way the server can update the information as soon as it is available, obtaining a real-time experience.

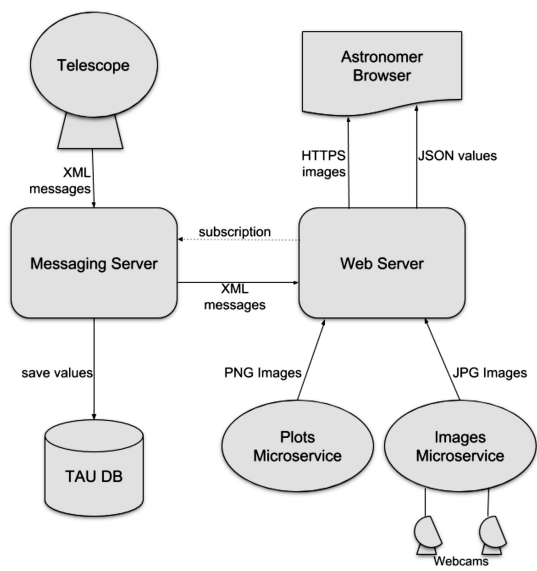


Figure 2. Data flow diagram.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

TOPCAT and Gaia

M. B. Taylor

H. H. Wills Physics Laboratory, Tyndall Avenue, University of Bristol, UK;
m.b.taylor@bristol.ac.uk

Abstract. TOPCAT, and its command line counterpart STILTS, are powerful tools for working with large source catalogs. ESA's *Gaia* mission, most recently with its second data release, is producing source catalogs of unprecedented quality for more than a billion sources. This paper presents some examples of how TOPCAT and STILTS can be used for analysis of *Gaia* data.

1. Introduction

TOPCAT (Taylor 2005) and STILTS (Taylor 2006) are respectively GUI and command-line analysis packages for working with tabular data in astronomy, and as such offer many facilities for manipulation of data such as source catalogs. The recent second data release from ESA's *Gaia* mission (Gaia Collaboration et al. 2018) has produced a source catalog of exceptional quality, and TOPCAT/STILTS are well-placed to provide analysis capabilities for exploitation of this data set. This paper discusses some of the features of the software most relevant for working with the *Gaia* DR2 catalog. Some have been added specifically with *Gaia* data in mind, but in most cases they are general purpose capabilities that are also suitable for use with other datasets.

2. Data Access

The primary access to the *Gaia* catalog is via Virtual Observatory (VO) protocols, provided from the main archive service at ESAC and a number of other data centers.

The most capable of these access protocols is TAP, the Table Access Protocol, which allows execution in the archive database of user-supplied SQL-like queries and retrieval of the resulting tables. TAP, while allowing extremely powerful queries to be performed, is a complex protocol stack which presents some challenges for the client software and user alike. TOPCAT provides the user with a GUI client for interacting with TAP services that integrates functions such as metadata browsing, query validation, table upload and query submission to make the use of TAP as straightforward as possible for the user, without obscuring the flexibility it offers (Taylor 2017). For simpler queries, a Cone Search client is also provided for retrieving source lists based on sky position alone. The *Gaia* catalog additionally contains non-scalar data for some rows, exposed using the VO DataLink protocol. At DR2 this array data is limited to epoch photometry of a relatively small number of known variable sources, but much more, including spectrophotometry, will be provided in future data releases. TOP-

CAT's *Activation Action* toolbox has been overhauled in recent versions to work with this array data.

Since these services follow the VO standards, no *Gaia*-specific code is required or implemented in TOPCAT to provide data access. This means that the same clients can be used for working with copies of the *Gaia* catalog in the main archive and elsewhere, as well as with other VO-compliant services. This standardization has benefits for both the implementer and user of the software.

The only truly *Gaia*-specific code in TOPCAT is a reader for the GBIN format used internally by the analysis consortium. This is a specialized capability of no interest to the general astronomy user, but it has proved valuable for DPAC members working with the data prior to catalog publication.

3. Expression Language

TOPCAT provides a powerful language for evaluating algebraic expressions to define new columns, row selections and plot coordinates. As well as standard arithmetic, trigonometric and astronomical operations, the library contains a number of astrometric functions:

- Propagation of astrometric parameters to earlier/later epoch, with or without errors and correlations
- Conversion of positions and velocities from astrometric parameters to Cartesian coordinates in ICRS, galactic or ecliptic coordinates
- Bayesian estimation of distances and uncertainties from parallax, using the expressions from Astraatmadja & Bailer-Jones (2016)

These are not exactly specific to *Gaia*, but they have been added as they are likely to be often needed when working with *Gaia* data, and they are specified and documented in a way that makes them easy to use in that context. For instance the following expression calculates the (U, V, W) components of velocity in the Galactic coordinate system (without adjusting for local standard of rest, and assuming that parallax error is low):

```
icrsToGal(astromUVW(ra, dec, pmra, pmdec,
radial_velocity, 1000./parallax, false))
```

The variable names here are `gaia_source` catalog column names, and the units are as supplied in the catalog.

4. Scalability

Gaia DR2 contains 1.7 billion sources, and investigating this dataset often requires working with large tables. TOPCAT is well-suited for interactive analysis, including flexible exploratory visualization, of tables (for instance selections from the full catalog) with the order of 10^6 – 10^7 rows. This regime is quite usable on modest hardware with no special data preparation, for instance data downloaded from external services, or loaded from local FITS or even CSV files. TOPCAT can be used with tables larger than this, but interactive performance may be poor or memory exhausted.

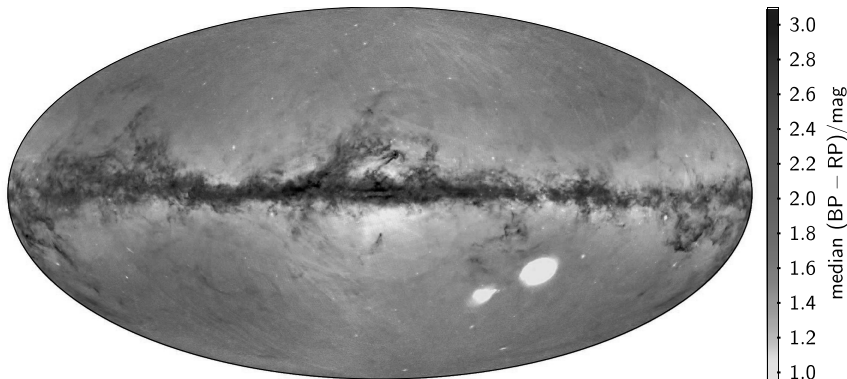


Figure 1. All-sky plot of $BP - RP$ color for 1.4 billion *Gaia* sources. The colors represent the median within each level 8 Halifax pixel.

STILTS on the other hand processes data for the most part in streaming mode, so can cope with arbitrarily large tables in fixed memory. This means that non-interactive calculations or preparation of graphics for the entire *Gaia* catalog is quite feasible. A set of all-sky weighted density maps using all 1.7 billion rows was prepared as follows:

1. download 61 000 (0.5 Tb) gzipped CSV files from ESAC (`wget`, 10 hours)
2. convert to 61 000 small FITS files (STILTS `tpipe`, 5 days)
3. convert to single 0.8 Tb column-oriented FITS file (STILTS `tpipe`, 12 hours)
4. aggregate into level-9 HEALPix map (STILTS `tskymap`, 45 minutes)
5. render plot to PDF or PNG (STILTS `plot2sky`, a few seconds)

In practice, steps 4 and 5 were iterated using TOPCAT visualization interactively on smaller datasets to archive the best results. An example is shown in Figure 1.

Note, this is not necessarily the best way to prepare such maps; executing the calculations near the data is in general more efficient (Taylor et al. 2016).

5. Visualization

TOPCAT has many visualization modes enabling highly configurable interactive exploration of high-dimensional data, and suitable for both large and small data sets. Special attention is given to providing comprehensible representations of large data sets — a simple scatter plot is not useful when there are many more points than pixels. Two plot types have been specifically introduced or enhanced for *Gaia* data: the *Sky Vector* plot displays proper motion vectors on the sky, and the *Sky/XY Correlation* plots show error ellipses based on the astrometric error and correlation quantities provided in the *Gaia* catalog; see Figure 2. The many non-*Gaia*-specific visualization options, too numerous to describe here, are however in most cases the core of TOPCAT's analysis capabilities for working with *Gaia* and non-*Gaia* data alike.

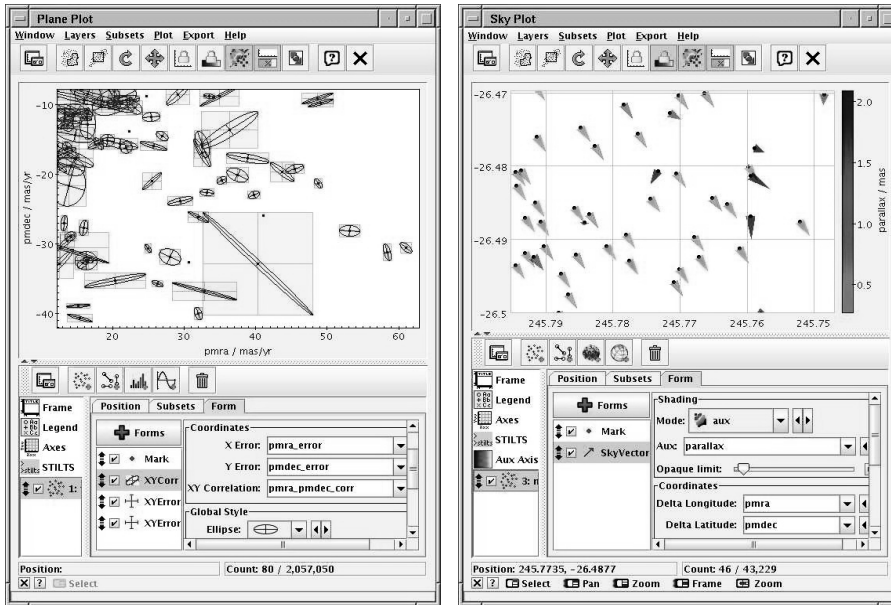


Figure 2. Interactive visualization of high-dimensional data in TOPCAT. The left hand figure displays proper motions with error ellipses derived from the Gaia `pmra_pmdec_corr` column, which show much more information than the simple `pmra_error/pmdec_error` error boxes. The right hand figure shows proper motion vectors by shape, and parallaxes by color. In each case, five dimensions are visualized.

Acknowledgments. This work has been primarily funded by the UK’s Science and Technology Facilities Council. It has made use of data from the European Space Agency (ESA) mission *Gaia* (<https://www.cosmos.esa.int/gaia>), processed by the *Gaia* Data Processing and Analysis Consortium (DPAC, <https://www.cosmos.esa.int/web/gaia/dpac/consortium>). Special thanks to the EU Horizon 2020 project ASTERICS for funding presentation of this work at ADASS 2018.

References

- Astraatmadja, T. L., & Bailer-Jones, C. A. L. 2016, *ApJ*, 833, 119. 1609.07369
 Gaia Collaboration, Brown, A. G. A., Vallenari, A., Prusti, T., de Bruijne, J. H. J., & al. 2018, *A&A*, 616, A1. 1804.09365
 Taylor, M. B. 2005, in ADASS XIV, edited by P. Shopbell, M. Britton, & R. Ebert, vol. 347 of ASP Conf. Ser., 29
 — 2006, in ADASS XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of ASP Conf. Ser., 666
 — 2017, in ADASS XXV, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of ASP Conf. Ser., 589
 Taylor, M. B., Mantelet, G., & Demleitner, M. 2016, in ADASS XXVI, ASP Conf. Ser. 1611. 09190

Session II

Machine Learning in Astronomy

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Deep Learning of Astronomical Features with Big Data

Maggie Lieu,¹ Deborah Baines,¹ Fabrizio Giordano,¹ Bruno Merin,¹
 Christophe Arviset,¹ Bruno Altieri,¹ Luca Conversi,¹ and Benoît Carry²

¹ESA, ESAC, Villanueva de la cañada, Madrid, Spain;
 maggie.lieu@sciops.esa.int

² Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Laboratoire
 Lagrange, Nice, France

Abstract. In Astronomy, there is a tendency to build machine learning codes for very specific object detection in images. The classification of asteroids and non-asteroids should be no different than the classification of asteroids, stars, galaxies, cosmic rays, ghosts or any other artefact found in astronomical data. In computer science, it is not uncommon for machine learning to train on hundreds of thousands of object categories, so why are we not there yet? I will talk about image classification with deep learning and how we can make use of existing tools such as the ESA science archive, ESAsky and citizen science to help realise the full potential of object detection and image classification in Astronomy.

1. Introduction

The big data revolution is upon us. Astronomy research is quickly becoming data-intensive with data volumes rapidly rising from gigabytes only a decade ago to petabytes today and even exabyte scales in the very near future. Traditional approaches to transferring, processing and analysing astronomical data is no longer sufficient to keep up with current data production rates in astronomy and all the while the amount of unexplored data in existing astronomical archives is also mounting up.

The development of automated tools are critical for efficient processing of the eminent wave of big data astronomy missions (see Ball & Brunner 2010, for a review) such as Euclid, LSST, Gaia and SKA, and to tackle data-mining of existing archives. Supervised learning is a sub-category of machine learning, a data-driven approach to automate the model building that maps data input layers to output layers by minimising the loss between output generated by the model and the ground truth labels (McCulloch & Pitts 1943). In neural networks, the intermediate layers consist of parameters that are optimised through training of data with known labels and the model is validated with data and labels that do not update the parameters of the model. The validation data determines when training should stop whilst also flagging any potential problems.

Currently supervised learning applied to large scale astronomy is hindered by the lack of data, the lack of ground truth labels and computational limitations required for training. After training however, the computational speed of a trained algorithm can be incredible fast because in general the underlying executed functions are very simple. In astronomy, many upcoming surveys will have huge data volume outputs however,

training data will not be available for supervised learning. For existing surveys, training datasets may be small and therefore overfitting to the data is a real concern and the ability to generalise on validation or testing data. Here we will propose solutions from obtaining training data, to dealing with limited computational resources, to ensure the full potential of existing datasets is met and to prepare us for future. Since in astronomy, a large fraction of our data is in the form of images, we therefore focus on machine learning applied to image data analysis. We will talk about such techniques in the context of upcoming big data missions and for data mining existing archives.

2. Image classification in astronomy

2.1. Training dataset

The training dataset (input data and labels) is an important aspect of machine learning since the performance of the trained algorithm and its ability to generalise is dependent on the training data. Models that are overfit to the training dataset will not generalise and will not perform well on validation or test data. There are three main causes of overfitting; firstly due to training the network for too long (which can be prevented by using validation to determine when to stop training), secondly if the training dataset is too small, and lastly if the network is too large. The latter two causes are closely related, the larger the network is, the more parameters to fit for and the more training data is required. Within the development of algorithms, data augmentation (rotation, scaling, cropping etc.) can help to artificially increase the sample size, also dropout (randomly dropping out layers), and batch norm (input normalisation) can also help with regularisation of the network.

The performance of network is also dependent on how representative the training dataset is of the testing data. Unrepresentative properties and biases inherent in the training dataset will propagate through to trained neural network (Tommasi et al. 2017). Whilst there are ways to mitigate such biases (e.g. weighting), the best way to ensure an unbiased neural network is by having a heterogeneous dataset that is representative of the evaluation data.

In some cases this may be difficult to obtain, in particular for upcoming surveys, where the data does not yet exist. Simulations can provide a large training dataset with labels, however realistic simulations that are representative of the evaluation data are hard to come by. Simulations can be improved by incorporating real data of existing and similar observations (see subsection 2.4). Instrument and observing condition specific biases inherent in the the training data (e.g. noise, PSF) can be mitigated by using a variation of observations sourced from different instruments and by folding in simulated instrumental properties. Furthermore, generative adversarial networks (GANs, Goodfellow et al. 2014), a reinforcement learning technique where two neural networks compete with each other, one a generator network that generates realistic data of interest and the other a discriminator network that tries to distinguish between the true and generated data, could be used to create realistic simulations (Ravanbakhsh et al. 2016; Mustafa et al. 2017; Schawinski et al. 2017).

In other cases, you may have a large volume of data but no corresponding labels for example the unexplored data within astronomical archives. Citizen science (see e.g. Lintott et al. 2008; Willett et al. 2013) is a pragmatic way to crowd source large numbers of volunteers to effectively analyse large amounts of data and obtain labels

(see subsection 2.6). The Zooniverse platform¹ provides an easy interface to create citizen science project and tools to analyse the data for free, but the process can take a while to gain sufficient participants. Alternatively Amazon's Mechanical Turk² offers a quick way to obtain large volunteers for a small cost.

2.2. Convolutional neural networks

Convolutional neural networks (CNNs LeCun et al. 1998) are deep machine learning networks, commonly used in analysing image data. In astronomy, CNNs are typically used for image recognition (Ackermann et al. 2018; Schaefer et al. 2018). They are a class of deep learning algorithms since they have multiple hidden (mostly convolutional and pooling) layers. The input data are a 3-dimensional array, such as images where the 3 dimensions correspond to the image width, height and channel depth (e.g. an RGB JPEG image has 3 channels). Each convolutional layer has a $n \times n$ convolutional kernel that multiplies with $n \times n$ areas of the input array and is then summed as it slides over it for a given padding (margin) and stride (pixel step). The number of convolutional layers used, the dimensionality of the kernels, the padding and the strides must be provided, whereas the values of the kernels are parameters to be learned from the data. Pooling layers reduce the dimensionality of inputs, for example a max pool layer would have a $m \times m$ pooling kernel that creates a new array with pixels that correspond to the maximum values of pixels in $m \times m$ areas on the input array. Again kernel dimensions, strides and padding need to be defined. Pooling layers allow the input data array to be reduced down to a single number that corresponds to the identified class and a probability.

2.3. Transfer learning

For deep neural networks like CNNs, determining the optimal number of layers, their types and other variables (strides etc) is a computationally heavy task and requires a lot of trial and error. It is quite common to base the architecture of new networks on existing high performing neural networks, however deep NNs typically have millions of parameters which could take weeks or months to train. Furthermore if the training dataset is too small, it is likely that the trained neural network will overfit to the training dataset and be unable to generalise in validation. In transfer learning (Pratt et al. 1991), the architecture of an existing high performing and extensively studied neural network is trained on existing large training datasets³ which may be completely unrelated to the problem of interest such as cats and dogs. The parameters of the initial layers are then fixed (frozen) and only the last layers are retrained to the training dataset of interest (e.g. stars and galaxies). Transfer learning removes the tedious process of designing a good architecture, and significantly reduces the number of parameters and hence training duration. Furthermore, the frozen parameters in the initial layers can be fine-tuned later on with more training, potentially increasing predicting accuracy by up to 15% (Khosravi et al. 2018). This works because the convolutional layers that correspond to lower level features (e.g. sharp edges and colours) are transferable to any dataset.

¹www.zooniverse.org

²<https://www.mturk.com/>

³see e.g. <http://deeplearning.net/datasets/>

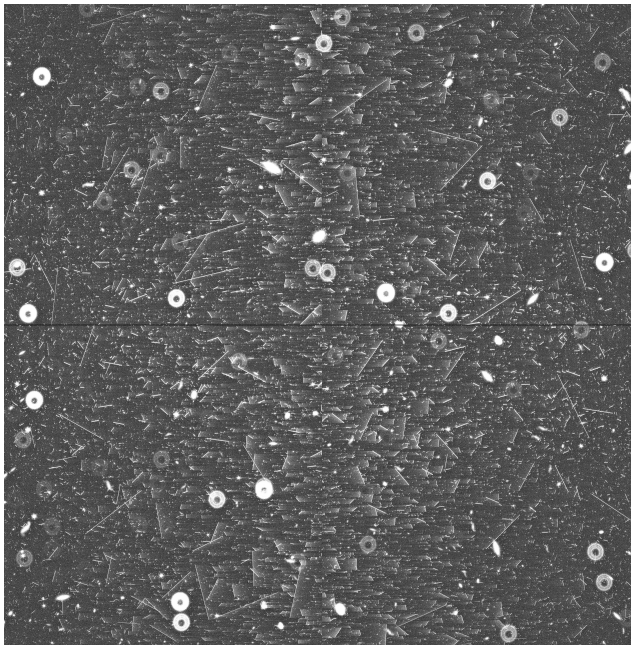


Figure 1. A simulation of one CCD on Euclid's visible imager (VIS).

In astronomy, the applications include removing glitches from gravitational wave data (George et al. 2018) and classifying mergers of galaxies (Ackermann et al. 2018).

2.4. Classifying solar system objects with Euclid

An example of where a training dataset is unavailable is ESA's upcoming space mission Euclid. Euclid will produce ~800GB of data daily for 6 years. It's main purpose is to measure accurate shapes of galaxies for weak gravitational lensing, however contaminants from asteroids (which appear elliptical due to their movement across the sky), stars (which may appear elliptical from the point spread function, PSF) and other effects (e.g. ghosts and cosmic rays) ideally need to be removed in real-time. Since Euclid is not yet launched there is no real data available for training. In Lieu et al. (2018), a training dataset is constructed from simulations based on cutouts of astronomical objects within NASA's Hubble archives and then convolved with the expected instrumental effects of Euclid (e.g. PSF, noise, charge transfer inefficiency). Figure 1 shows an example of the simulated data. As CNNs only work for images containing a single object, the training dataset they use comprises of randomly sized image cutouts centred on a balanced dataset of asteroids stars, galaxies and cosmic rays. Using the MobileNet_v1_1.0_224 architecture (Howard et al. 2017) trained on the ImageNet 2012⁴ dataset (consisting of 14 million images belonging to 200,000 different classes), they retrain a new top layer to represent the new classes on a laptop without GPU. With

⁴<http://www.image-net.org/challenges/LSVRC/2012/>

less than 2 hours of training they are able to achieve an accuracy of 94% of binary classification of asteroids and 83% when using all 4 categories. Once trained any new classifications are made in less than a second, are highly parallelisable and unbiased with respect to asteroid magnitude and speed. The trained neural network could be used for real-time detection of solar system objects, allowing for immediate follow up observations with ground based telescopes.

2.5. Object detection

CNNs work well on images of single objects, however in many cases it may not be possible to obtain such images, in particular in the presence of multiple overlapping objects with many different labels. In object detection we aim to locate and classify objects in images (see Zhao et al. 2018, for a review). There are 2 main methods to do this:

Firstly region proposal networks such as regional convolutional neural network (R-CNN, Girshick et al. 2014), Fast R-CNN (Girshick 2015) and Faster R-CNN (Ren et al. 2015). R-CNNs are even deep CNNs that require not only true classes of objects but also parameters that define a bounding box that encompasses the object. They combine CNNs with bounding box regression in order to not only classify objects but to also detect them, allowing the possibility of multi-object classification and for a variable number of objects in each image. The bounding box regression returns a position, height and width of the box that encloses the class, and optimization is performed to minimize the overlap between the predicted box and the truth. R-CNNs are effectively 2 neural networks, the first proposes regions of interest using selective search to group similar areas of an image. These regions are fed individually into a CNN to extract features. The second network is a support vector machine (SVM) that is used to determine the presence of a class in each proposed region by separating the feature space and also predict the bounding box of the objects. These object detection networks are very accurate however computationally very expensive.

The other methods are single shot object detectors such as single shot multibox detector (SSD, Liu et al. 2015) and YOLO (Redmon et al. 2015; Redmon & Farhadi 2016). SSD requires only a single network but instead looks for objects in pre-defined boxes (anchors)⁵ of given scales and aspect ratios (since humans tend to appear elongated whereas bubbles tend to appear circular). The network is a regular CNN where only the latter layers are used for object search where the output dimensions are significantly smaller in comparison to the initial layers. Here fewer anchors are required and it is also much faster to train because there is no need for proposals. If an object is present within an anchor, the SSD will optimize the difference between the pre-defined anchor and the nearest true bounding box and also predict a class probability. Furthermore SSD makes use of hard negative mining to learn the regions that do not overlap with any classes which further increases the speed and stability in training. SSD requires the scales, aspect-ratios and the number of predefined boxes to be specified, the ratio of hard negatives to positives (regions containing an object) and the maximum number of objects per image. SSDs are incredibly fast but trade off on accuracy. They don't perform well for crowded objects or very small objects since it works on the low resolution outputs of the latter layers. SSDs also suffer from multiple detections of the

⁵We note that faster R-CNN also uses anchors.

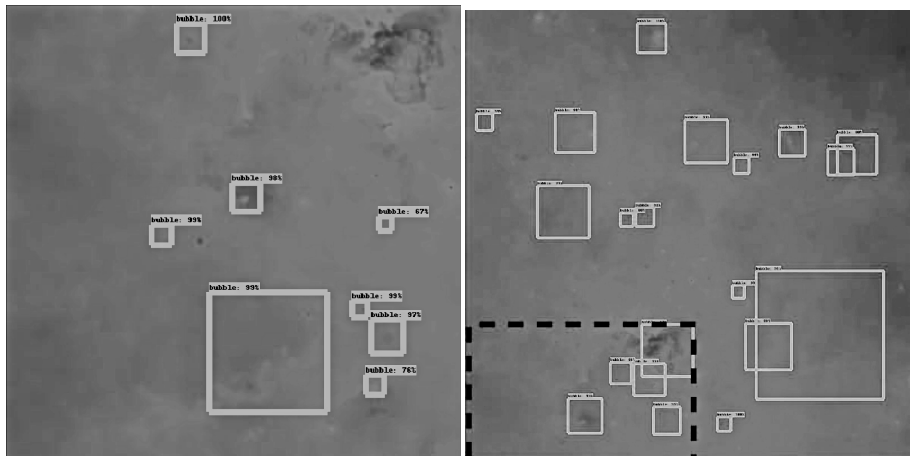


Figure 2. Far-infrared images from combined Herschel PACS 70 and 160 microns. The green boxes show the bubbles detected with over 50% probability by the trained SSD and the probability of the classification. *Left*: 600x600 image input. *Right*: 1200x1200 image input. The dashed line corresponds to the area covered by the left image.

same object, however this can be reduced by reducing the allowed area of overlapping between boxes with the same classifications.

2.6. Detecting space bubbles in the Herschel archives

The ESA science data archives are a gold mine for data mining with machine learning. One example of such unexplored data is the Herschel science archive where thousands of interstellar bubbles remain unidentified. These regions are typically caused by interactions between the stellar winds of young hot stars and their surrounding gas. They are therefore associated with star forming regions. The bubbles are easily identifiable by eye but it would be a very tedious task, and un-replicable, since humans are known to be slow and inconsistent. Instead we use object detection to automate the localisation of these bubbles using combined images from Herschel PACS survey at 70 and 160 microns (far-infrared) on the galactic plane. We rely on bubble detection catalogue from the Zooniverse citizen science project - The Milkyway Project (Simpson et al. 2012) as labels to train an SSD. The classifications and enclosing box parameters are identified by over 35,000 volunteers using GLIMSE and MIPS GAL images from NASA's Spitzer telescope (mid-infrared). It consists of 3745 bubbles, where each bubble is identified by at least 5 individuals. The input Herschel images are cropped to 1200×1200 pixels in size and then further rescaled to 300×300 pixels. Using transfer learning we train on the SSD_MobileNet architecture that is pre-trained on the Microsoft coco⁶ dataset as opposed to ImageNet 2012 data since we need additional information about bounding boxes for each object. MS coco consists of 330,000 images with 1.5M objects in 80 object categories. We split the labelled dataset into 90% training and 10% valida-

⁶<http://cocodataset.org>

tion. With a max anchor value of 150, we achieve a training rate of 2 seconds per step. Training completes in approximately 1 day using a laptop without using GPU, which is determined by a plateau in the validation data loss and 35k steps. Again once trained, any further classification takes only second so therefore we apply the algorithm to all the remaining Herschel images which are not covered by the Spitzer data.

Quantification of the performance of object detection is not trivial, the industry standard is to use the mean average precision (mAP) metric at e.g 0.5 intersection-over-Union (IoU), where

$$\text{IoU} = \frac{\text{number of pixels in the intersection between true and predicted box}}{\text{number of pixels in the union between true and predicted box}} \quad (1)$$

We obtain a low mAP@0.5IoU score of 0.07 for the 1200×1200 images due to many undetected small objects, however once trained we can rescale any input image regardless of size to 300×300 to detect star forming bubbles on various scales as we show in Figure 2. The 600×600 image detects many of the missing objects in the 1200×1200 image. To obtain a high mAP score requires predicted boxes to be aligned in position and scale, however the reliability of our labels is unclear since they originate from mid-infrared wavelengths rather than the far-infrared used in the training, and furthermore Beaumont et al. (2014) found that 10-30% of the objects in the Milkyway Project are interlopers. More work needs to be done in the future to help quantify object detection in a useful manner including combining different scale images, dithered images to help detect objects close to the image edge and removal of duplicate objects without losing objects with large amounts of overlap.

3. ESAsky implementation

For demonstration purposes we built a tool to visualise the detection boxes in ESA's ESAsky python widget - PyESAsky (Figure 3). The tool allow users to upload .csv files containing the positions of the bounding box parameters and visualise them overlaid on any of the many wavelength layers available with user defined width and colour. Visually it is clear that the SSD has identified correctly the bubbles and there is good agreement with the citizen science identifications. It is also possible to filter objects with low detection scores from the SSD MobileNet output and to flag publications associated to objects. This combined with the ease to quickly alternate between the multi-wavelength data available is a powerful tool for scientific exploration. Note that some objects detected by the SSD, were not detected by the citizens and vice versa. It is important to remember that the former were classified on F-IR images and the latter M-IR images. Nonetheless, star-forming regions are star-forming regions regardless of which wavelengths they are observed in. Equally, the same holds for any astronomical objects (stars, galaxies, supernovae etc.) and instrumental features (ghosts, bad pixels etc.). By exploiting our data archives and the full spectrum and variety of data available, we have will be able to maximise our scientific outputs, and find new exciting sources to investigate in more detail.

4. Conclusions

Machine learning is quickly becoming a necessity with the current growth of astronomical data acquisition. Supervised machine learning requires large quantities of data

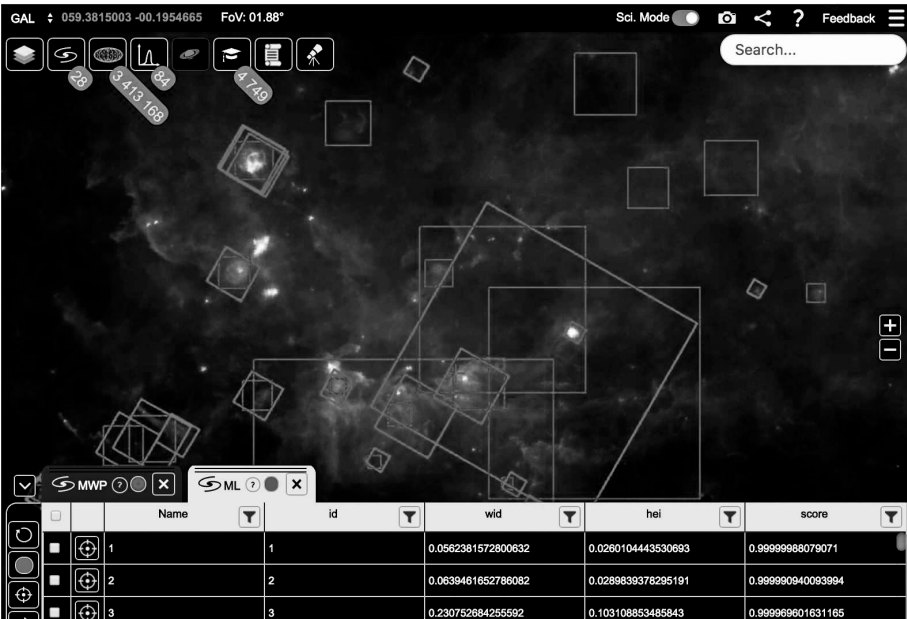


Figure 3. PyESAsky interface demonstrating the uploaded bounding boxes of star forming bubbles detected in the M-IR by citizen science (green) and those detected by the SSD MobileNet (red) on Herschel PACS HIGAL images.

which we have in our archives and will continue to obtain through upcoming astronomical surveys, and ground truth data which is often not easy to obtain. Furthermore, the advancement of deep learning networks are computationally expensive and can easily take months to train. Here we have discussed 2 applications of machine learning on astronomical image data and how to make use of the synergies between different instruments and wavelengths.

First, dealing with the upcoming big data era, where there is a need for real-time quick analysis however a training dataset is not yet available. We introduced convolutional neural networks to classify single objects in images from ESA's upcoming Euclid mission. The training data were simulated based on real observations from Hubble space telescope combined with the Euclid pipeline, and since they are simulations, the ground truth is known. This approach works when it is plausible to obtain cutouts of single objects and once trained it is extremely fast, accurate and unbiased.

Second, we demonstrate the power of merging the two different wavelengths and instrument data for classification when exploiting existing archives. We use Spitzer (M-IR) labels of bubble regions combined with Herschel (F-IR) images to train an object detection network. The labels are obtained through citizen science which reduces the time needed for obtaining a large quantities of labels compared to a single scientist classifier. The object detection is more expensive to train but is preferable over the convolutional neural network when cutouts of single objects are not available or in cases with multiple overlapping objects. In both applications transfer learning with big datasets were used to reduce the number of training images required and computational strain. In the future, we are looking into the feasibility to integrate citizen science directly into ESAsky by allowing user defined classification of both astronomical objects and features. The data collected could be fed into further improving machine learning algorithms for object detection and classification within ESAsky. The large volume multi-wavelength resources could help with the generalisation of object detection and classification algorithms and which could play an important role in any real-time analysis of large data missions in the near future. Cross-matching detections, known publications and visual inspection of images across the multi-wavelength, multi-instrument interface of ESAsky has enormous potential for uncovering unexplored and peculiar sources for science in the near future.

Acknowledgments. ML acknowledges a ESA Research Fellowship at the European Space Astronomy Centre (ESAC) in Madrid, Spain. This publication uses data generated via the Zooniverse.org platform, development of which is funded by generous support, including a Global Impact Award from Google, and by a grant from the Alfred P. Sloan Foundation.

References

- Ackermann, S., Schawinski, K., Zhang, C., Weigel, A. K., & Turp, M. D. 2018, MNRAS, 479, 415. 1805.10289
- Ball, N. M., & Brunner, R. J. 2010, International Journal of Modern Physics D, 19, 1049. 0906.2173
- Baumont, C. N., Goodman, A. A., Kendrew, S., Williams, J. P., & Simpson, R. 2014, ApJS, 214, 3. 1406.2692
- George, D., Shen, H., & Huerta, E. A. 2018, Phys.Rev.D, 97, 101501. 1706.07446
- Girshick, R. 2015, ArXiv e-prints. 1504.08083

- Girshick, R., Donahue, J., Darrell, T., & Malik, J. 2014, in Proceedings of the IEEE conference on computer vision and pattern recognition, 580
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. 2014, ArXiv e-prints. 1406.2661
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. 2017, ArXiv e-prints. 1704.04861
- Khosravi, P., Kazemi, E., Imielinski, M., Elemento, O., & Hajirasouliha, I. 2018, EBioMedicine, 27, 317 . URL <http://www.sciencedirect.com/science/article/pii/S2352396417305078>
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, Proceedings of the IEEE, 86, 2278
- Lieu, M., Conversi, L., Altieri, B., & Carry, B. 2018, ArXiv e-prints. 1807.10912
- Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., Murray, P., & Vandenberg, J. 2008, MNRAS, 389, 1179. 0804.4483
- Liu, W., Angelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. 2015, ArXiv e-prints. 1512.02325
- McCulloch, W. S., & Pitts, W. 1943, The bulletin of mathematical biophysics, 5, 115. URL <https://doi.org/10.1007/BF02478259>
- Mustafa, M., Bard, D., Bhimji, W., Lukić, Z., Al-Rfou, R., & Kratochvil, J. 2017, ArXiv e-prints. 1706.02390
- Pratt, L. Y., Mostow, J., & Kamm, C. A. 1991
- Ravanbakhsh, S., Lanusse, F., Mandelbaum, R., Schneider, J., & Poczos, B. 2016, ArXiv e-prints. 1609.05796
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2015, ArXiv e-prints. 1506.02640
- Redmon, J., & Farhadi, A. 2016, ArXiv e-prints. 1612.08242
- Ren, S., He, K., Girshick, R., & Sun, J. 2015, ArXiv e-prints. 1506.01497
- Schaefer, C., Geiger, M., Kuntzer, T., & Kneib, J.-P. 2018, A&A, 611, A2. 1705.07132
- Schawinski, K., Zhang, C., Zhang, H., Fowler, L., & Santhanam, G. K. 2017, MNRAS, 467, L110. 1702.00403
- Simpson, R. J., Povich, M. S., Kendrew, S., Lintott, C. J., Bressert, E., Arvidsson, K., Cyganowski, C., Maddison, S., Schawinski, K., Sherman, R., Smith, A. M., & Wolf-Chase, G. 2012, MNRAS, 424, 2442. 1201.6357
- Tommasi, T., Patricia, N., Caputo, B., & Tuytelaars, T. 2017, in Domain Adaptation in Computer Vision Applications (Springer), 37–55
- Willett, K. W., et al. 2013, MNRAS, 435, 2835. 1308.3496
- Zhao, Z.-Q., Zheng, P., Xu, S.-t., & Wu, X. 2018, ArXiv e-prints. 1807.05511



Isabelle Joncour, Kai Polsterer and Maggie Lieu at the final box lunch (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Automatic Classification of Transiting Planet Candidates using Deep Learning

M. Ansdell,¹ Y. Ioannou,² H. P. Osborn,³ M. Sasdelli,⁴ J. C. Smith,^{5,6} D. Caldwell,^{5,6} J. M. Jenkins,⁵ C. Räissi,⁷ and D. Angerhausen⁸

¹*Department of Astronomy, University of California at Berkeley, CA, USA;*
ansdell@berkeley.edu

²*Machine Intelligence Lab, University of Cambridge, UK*

³*Laboratoire d'Astrophysique de Marseille, France*

⁴*Australian Institute for Machine Learning, University of Adelaide, Australia*

⁵*NASA Ames Research Center, Moffet Field, CA, USA*

⁶*SETI Institute, Mountain View, CA, USA*

⁷*Institut national de recherche en informatique et en automatique, France*

⁸*Center for Space and Habitability, University of Bern, Switzerland*

Abstract. Space-based missions such as *Kepler*, and now *TESS*, provide large datasets that must be analyzed efficiently and systematically. Shallue & Vanderburg (2018) recently used state-of-the-art deep learning models to successfully classify *Kepler* transit signals as either exoplanets or false positives. We expand upon that work by including additional “scientific domain knowledge” into the network architecture and input representations to significantly increase overall model performance to 97.5% accuracy and 98.0% average precision. Notably, we achieve 15–20% gains in recall for the lowest signal-to-noise transits that can correspond to rocky planets in the habitable zone. This work illustrates the importance of including expert domain knowledge in even state-of-the-art deep learning models when applying them to scientific research problems that seek to identify weak signals in noisy data.

1. Introduction

Exoplanet science is no longer data limited: the *Kepler* (Borucki 2016) and *TESS* space missions produce copious amounts of data that need to be processed quickly and systematically in order to enable efficient follow-up observations and yield reliable statistics on exoplanet occurrence rates. These observatories work by measuring the brightness of target stars as a function of time, producing a flux time series known as a light curve; exoplanets are identified when they transit in front of the star, causing a drop in the observed brightness. However, exoplanet signals are small compared to instrumental noise/systematics as well as inherent stellar variation also present in the data. Additionally, false-positive planet signals, such as those due to eclipsing binaries (EBs) and background eclipsing binaries (BEBs), need to be reliably culled. Thus

manual vetting by human operators can often result in biased or incomplete samples. Machine learning offers a new approach capable of rapidly and reliably identifying exoplanet transits. Shallue & Vanderburg (2018) successfully developed a deep convolutional neural network for automatically classifying candidate exoplanet transits in *Kepler* data, however improvements could be made with the inclusion of additional “scientific domain knowledge”—i.e., the information, insight, or intuition relevant to a specific problem that a domain expert can provide. Here we present results that investigated these possibilities. All code and data used in this work is publicly available.¹

2. Data & Labels

We use the Q1–Q17 *Kepler* Data Release 24 (DR24) light curves. Each light curve contains one or more “threshold crossing events” (TCEs) identified by the *Kepler* Science Processing Pipeline (Jenkins et al. 2010); each TCE is a potential exoplanet transit event, however most TCEs will be false-positive signals, sometimes caused by astrophysical phenomena such as EBs or BEBs, but often by instrumental noise or other spurious events. Following Shallue & Vanderburg (2018), we flatten the light curves by dividing by an iteratively fitted basis spline (see Fig. 3 in Vanderburg & Johnson 2014), then create “global” and “local” views of each phase-folded TCE (see Fig. 3 in Shallue & Vanderburg 2018). Both views are scaled so that the continuum is at 0 and the maximum transit depth is at -1 . The global view encapsulates the full view of the phase-folded light curve (e.g., including secondary transits of EBs) at the cost of long-period TCEs having poorly sampled transits. The local view, which depends on the transit duration, then provides a more detailed view of the primary transit shape.

We also use centroid curves, which are the time-series of the pixel position of the center of light within a photometric aperture. Centroid curves are particularly useful for identifying BEBs, as centroid positions will shift if both the BEB and target star are contained within the same photometric aperture. We use the x and y pixel coordinates of the centroid to compute the absolute magnitude ($r = \sqrt{x^2 + y^2}$) of the centroid displacement. We then follow the same process as the light curves for smoothing, phase-folding, and translating into local and global views. However, rather than normalizing the centroid curve to the maximum transit depth, we subtract the median and divide by the standard deviation, where these values are calculated out-of-transit and across the entire training dataset (this standard practice is called “normalization” in machine learning). Moreover, we normalize the standard deviation of the centroid curves by that of the light curves, which ensures that TCEs with no significant centroid shifts show flat lines with noise signal strengths similar to that of the light curves (and thus do not dominate the signal strengths). Finally, we use the updated *Kepler* DR25 catalog (Mathur et al. 2017) to obtain intrinsic stellar parameters, namely effective temperature (T_{eff}), surface gravity ($\log g$), metallicity ($[\text{Fe}/\text{H}]$), radius (R_\star), mass (M_\star), and density (ρ_\star). These stellar parameters are normalized as for the centroids.

We use the same labels as Shallue & Vanderburg (2018), taken from the *Kepler* DR24 TCE Table available on the NASA Exoplanet Archive. The **av_training_set** column contains the labels used to train the *Autovetter* (McCauliff et al. 2015) and primarily come from human-vetted KOIs assembled from multiple papers (e.g. Batalha

¹<http://gitlab.com/frontierdevelopmentlab/exoplanets>

et al. 2013). The **av_training_set** column has four possible values: planet candidate (PC), astrophysical false positive (AFP), non-transiting phenomenon (NTP), and unknown (UNK). Following Shallue & Vanderburg (2018), we ignore the UNK TCEs (4,630 entries) and then bin the remaining labels as “planet” (PC; 3,600 entries) or “false positive” (AFP + NTP; 12,137 entries). We then divide the TCEs into training (80%), validation (10%), and test (10%) sets using the same random seed as Shallue & Vanderburg (2018) to preserve comparability. We use the validation set to tune hyper-parameters and the test set for our final model performance results.

3. Models

The baseline model is **Astronet**, the deep convolution neural network developed by Shallue & Vanderburg (2018). In short, the **Astronet** model architecture has two disjoint one-dimensional convolutional columns (one for the global view and one for the local view) with max pooling, the results of which are concatenated and then fed into a series of fully connected layers ending in a sigmoid function that produces an output in the range (0,1) that loosely represents the likeliness of a given TCE being a true planet transit (1) or false positive (0). For model training, **Astronet** uses the Adam optimization algorithm (Kingma & Ba 2014) to minimize the cross-entropy error function. During training, data are augmented by applying time inversions to the input light curves with a 50% chance. Our **Astronet** performance results are consistent with those reported in Shallue & Vanderburg (2018); for example, we find an accuracy of 0.958 compared to their 0.960 value.

Here we use scientific domain knowledge to add several features to our baseline **Astronet** model architecture and input representations in an effort to increase model performance. This modified model, which we call **Exonet**, inputs our analogous global and local views of the centroid curves as second channels of the disjoint convolutional columns used for the light curves to help the model learn the connections between the shapes of the light curves and centroid curves, which can be useful for identifying false positives, in particular BEBs. We also concatenate the stellar parameters to the flattened outputs of the convolutional layers directly before feeding them into the shared fully connected layers. We add this information because stellar parameters are likely correlated with classification, for example giant stars with large radii are far more likely to host stellar eclipses than planetary transits (which would be undetectable). **Astronet** augments the data by randomly flipping the time axis of half the input light curves during training; we adopt this data augmentation technique, also applying the time-axis flip to the associated centroid curves. Because we found that **Astronet** suffers from model over-fitting, we apply an additional data augmentation technique during training to mimic measurement uncertainties in the flux measurements: namely, we add random Gaussian noise to each input light curve, where the standard deviation is randomly chosen from a uniform distribution between 0 and 1.

4. Results & Conclusions

To assess model performance, we use three key metrics: accuracy, average precision, and precision-recall curves. Using model ensembling on the test set, we achieve 97.5% accuracy and 98.0% average precision, corresponding to increases of 1.7% and 2.5%,

respectively, over *Astronet*. Figure 1 then shows precision-recall curves with each component of scientific domain knowledge added individually to illustrate their separate contributions to improving model performance; for this, we use k -fold cross-validation on the combined training and validation sets with $k = 5$. Figure 1 also shows the precision and recall as a function of a measure of the signal-to-noise of the candidate transits—the so-called “multiple event statistic” (MES; Jenkins et al. 2002) that the *Kepler* pipeline reports with each TCE. Notably, *Exonet* shows 15–20% increases in recall for low-MES transits that often correspond to Earth-sized planets, some of which are in the habitable zone.

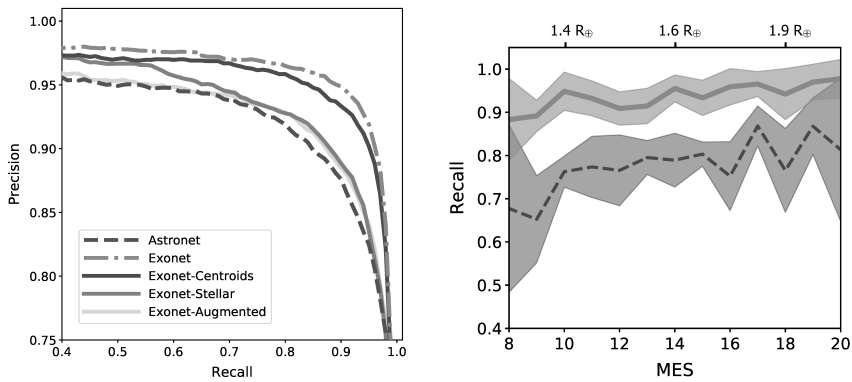


Figure 1. Comparisons between *Astronet* and *Exonet* model performance. *Left*: precision-recall curves showing the individual contributions of scientific domain knowledge to model performance. The centroid curves provide the biggest gains in model performance, while stellar parameters also make a significant impact. Data augmentation does not greatly increase model performance, but rather the main benefit is to alleviate model over-fitting. *Right*: Recall as a function of MES (signal-to-noise of the candidate transit), illustrating that the gains in performance by *Exonet* can be most significant for Earth-sized planets.

In summary, this work demonstrates the importance of including domain knowledge in even state-of-the-art machine learning models when applying them to scientific research problems that seek to identify weak signals in noisy data. This classification tool will be especially useful for upcoming space-based photometry missions focused on finding small planets, such as *TESS* and *PLATO*.

References

- Batalha, N. M., et al. 2013, *ApJS*, 204, 24. 1202.5852
 Borucki, W. J. 2016, *Reports on Progress in Physics*, 79, 036901
 Jenkins, J. M., Caldwell, D. A., & Borucki, W. J. 2002, *ApJ*, 564, 495
 Jenkins, J. M., et al. 2010, *ApJ*, 713, L87. 1001.0258
 Kingma, D. P., & Ba, J. 2014, *ArXiv e-prints*. 1412.6980
 Mathur, S., et al. 2017, *The Astrophysical Journal Supplement Series*, 229, 30
 McCauliff, S. D., et al. 2015, *The Astrophysical Journal*, 806, 6
 Shallue, C. J., & Vanderburg, A. 2018, *AJ*, 155, 94. 1712.05044
 Vanderburg, A., & Johnson, J. A. 2014, *PASP*, 126, 948. 1408.3853

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Acceleration of Non-Linear Minimization with PyTorch

Bojan Nikolic

Astrophysics Group, Cavendish Lab., University of Cambridge, Cambridge
CB3 0HE, UK; b.nikolic@mrao.cam.ac.uk

Abstract. Minimization (or, equivalently, maximization) of non-linear functions is a widespread tool in astronomy, e.g., maximum likelihood or maximum a-posteriori estimates of model parameters. Training of machine learning models can also be expressed as a minimization problem (although with some idiosyncrasies). This similarity opens the possibility of re-purposing machine learning software for general minimization problems in science.

I show that PyTorch, a software framework intended primarily for training of neural networks, can easily be applied to general function minimization in science. I demonstrate this with an example inverse problem, the Out-of-Focus Holography technique for measuring telescope surfaces, where a improvement in time-to-solution of around 300 times is achieved with respect to a conventional NumPy implementation. The software engineering effort needed to achieve this speed is modest, and readability and maintainability are largely unaffected.

5. Introduction

The unconstrained minimization problem is posed as:

$$\operatorname{argmin}_{\vec{x} \in R^n} f(\vec{x}) \quad (1)$$

where f is the single-valued function to be minimized and its n parameters are expressed as if they were components of vector \vec{x} . Most non-linear minimization algorithms require the gradient of the function to be minimized (Nocedal & Wright 2006), i.e. $\nabla f = \frac{\partial f}{\partial x_i}$.

In a wide class of problems it is expensive (in terms of computational resources) to compute $f(\vec{x})$ and ∇f meaning the minimization as a whole is expensive. Additionally, there are time-sensitive applications, e.g., in control systems, where the latency between observation and the solution is important even if the overall expense is not. For these reasons it is desirable to accelerate minimization as a whole and in particular the evaluation of $f(\vec{x})$ and ∇f . At the same time, the function to be minimized and its gradient can be very complex (and subject to evolution over time), yet it is *essential* that they are implemented correctly. For this reasons, extensive by-hand optimization of the code of their implementation is undesirable as this is typically inflexible and error prone.

Training of neural networks can also be expressed as a minimization problem. Although the algorithms used for this minimization are sometimes specific (to take into account that only a part of the available training set is considered at a time), they usually

NumPy:	PyTorch:
<pre>def gauss(x0, y0, amp, sigma, rho, diff, a): dx=a[... ,0]-x0 dy= a[... ,1]-y0 r=numpy.hypot(dx, dy) R2= (r**2 + rho*(dx*dy)+ diff*(dx**2-dy**2)) E=numpy.exp(-1.0/ (2*sigma**2)*R2) return amp*E</pre>	<pre>import torch as T def hypot(x, y): return T.sqrt(x**2 + y**2) def gauss(x0, y0, amp, sigma, rho, diff, a): dx=a[... ,0]-x0 dy= a[... ,1]-y0 r=hypot(dx, dy) R2=(r**2 + 00(rho*(dx*dy))+ 00(diff*(dx**2-dy**2))) E=T.exp(-R2/(2*sigma**2)) return 00(amp*E)</pre>

Figure 2. Comparison of NumPy and PyTorch implementations of a function that models the Gaussian illumination (or apodization) of the aperture plane. The input parameters to the Gaussian function are it position (x0, y0), amplitude (amp), shape (sigma, rho and diff), and the raster on which it is to be evaluated (a). Function 00 handles to optional offload onto GPUs.

involve frequent evaluation of potentially expensive $f(\vec{x})$ and ∇f . Rapid adoption of neural networks in information technology systems has lead to significant investment into software to support their training, including PyTorch (Paszke et al. 2017). The new software packages developed in this area emphasize both efficiency and ease of use.

In this paper I show that PyTorch can easily be used for general minimization problems in science and that its qualities of ease of use and efficiency are maintained.

6. Why PyTorch ?

PyTorch² has four key features which make it particularly suitable for accelerating non-linear function minimization in science and engineering: **(1)** The interface is in Python, modelled after the widely used NumPy (Oliphant 2007) library, and introduces very few restrictions on the programmer. **(2)** It supports automatic reverse-mode differentiation for very efficient computation of ∇f without user intervention. **(3)** Easy-to-use offloading of operations onto Graphical Processing Units (GPUs) for efficient computation of data-parallel operations. **(4)** It is available as high-quality open-source distribution

These features combine into a programming environment that is familiar to engineers and scientists while enabling accurate and high performance minimization.

²<https://pytorch.org/>

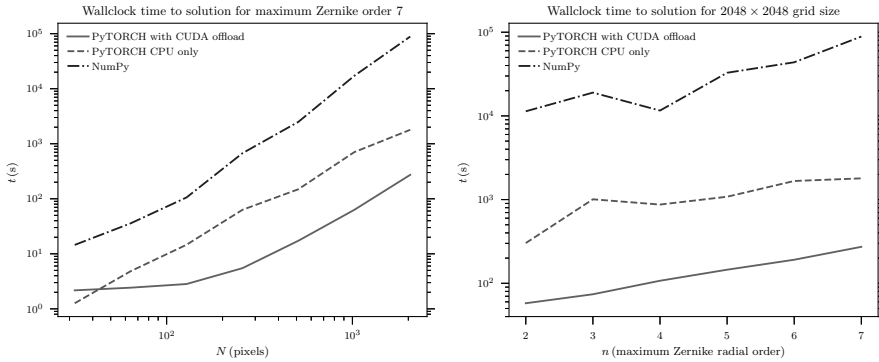


Figure 3. Time to solution as function of the grid size (left) and maximum Zernike polynomial order (right) used for modelling the telescope optics. All measurements on a dual-socket Intel Xeon CPU E5-2630 with dual NVidia Tesla K20c GPUs. **NumPy**: plain CPU-only NumPy implementation; **PyTorch CPU only**: implementation using PyTorch but without using offloading onto the GPU; **PyTorch with CUDA offload**: implementation using PyTorch and selecting offloading onto GPUs using the PyTorch CUDA module.

7. Example application: Maximum Likelihood Phase Retrieval

I illustrate the application of PyTorch to function minimization in science by applying it to the phase retrieval problem. Phase retrieval is the derivation of the phase of an oscillatory field (electromagnetic or particle-wave) from far-field measurements of its power only. It is a common technique used in demanding scientific imaging applications. I show here an application to a maximum-likelihood, model-based, phase retrieval in radio astronomy used to measure and optimize some of the largest single-dish radio telescopes (Nikolic et al. 2007). This specific technique is called Out-Of-Focus (OOF) holography: out-of-focus because the optical system is intentionally defocused to introduce a known phase change (‘phase diversity’) into it; holography because, as in Gabor’s sense, both the amplitude and the phase of the field are ultimately obtained.

In the original paper (Nikolic et al. 2007), normally distributed independent measurement error was assumed leading to a least-squares formulation. Here we consider the more complex extension to a Cauchy likelihood function:

$$P(y_i|\hat{y}_i) = \frac{1}{\pi\gamma} \frac{\gamma^2}{(y_i - \hat{y}_i)^2 + \gamma^2} \quad (2)$$

where γ represents the noise of measurements. This distribution is physically motivated for the radio astronomy phase-retrieval because in the very high dynamic range regime in which the input data are observed, errors such as pointing errors or inaccurate atmospheric subtraction can lead to non-Gaussian errors with wide tails. Similar consideration apply in a range of maximum-likelihood problems where there is a possibility for a few measurements with very high error, due to e.g., glitches in read out systems, external unpredictable events (e.g., cosmic ray hits), etc.

7.1. Implementation

Transforming a NumPy implementation into a PyTorch implementation took around four hours of programming time and produced a model of comparable complexity and readability. Several reasons were observed for the relatively low amount of work: firstly, only parts of the models that are functions of the model parameters need to be translated in PyTorch, meaning some relatively complex parts such as calculating the rasterized Zernike polynomials did not need translation at all. Secondly, most NumPy functions have direct counterparts in PyTorch; those that did not have counterparts were all written in NumPy themselves (not in a C extension) and could easily be re-implemented in PyTorch with reference to their source code. A typical translation into PyTorch is illustrated in Figure 2.

7.2. Performance Measurement

Relative performance of the NumPy and PyTorch variants were measured on a dedicated Dell PowerEdge server with dual socket Intel Xeon E5-2630 CPUs and dual NVIDIA Tesla K20c GPUs. Measurement was made as function of:

1. Number of model parameters, represented by the maximum order of Zernike polynomials used, n . The actual number of polynomials up to order n is $n(n + 3)/2 + 1$, so for example if Zernike polynomials of to order $n = 8$ are used there are 45 parameters to be optimized.
2. Computational cost of the model calculation, represented number of pixels N in each dimension of the grid.

Simulated measurements (including simulated noise) were used as an input into the phase retrieval and it was found that the different implementations converged to the same result up to the tolerances specified to the BFGS algorithm.

It can be seen that the PyTorch implementation is far faster in all configuration, and that the GPU-offloaded execution is faster than the CPU-only execution above grid size $N \geq 64$. For intermediate and large grids ($N > 256$) the PyTorch implementation running on CPUs is approximately an order of magnitude faster than the NumPy implementation, while the GPU-offloaded execution is an order of magnitude faster still.

Acknowledgments. I am pleased to acknowledge the support of the EC ASTER-ICS Project (Grant Agreement no. 653477).

References

- Nikolic, B., Hills, R. E., & Richer, J. S. 2007, A&A, 465, 679. [astro-ph/0612241](#)
 Nocedal, J., & Wright, S. J. 2006, Numerical Optimization (New York: Springer), 2nd ed.
 Oliphant, T. E. 2007, Computing in Science Engineering, 9, 10
 Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. 2017

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Feature Selection for Better Spectral Characterization or: How I Learned to Start Worrying and Love Ensembles

Sankalp Gilda

University of Florida, Gainesville, FL, U.S.A; s.gilda@ufl.edu

Abstract. An ever-looming threat to astronomical applications of machine learning is the danger of over-fitting data, also known as the ‘curse of dimensionality.’ This occurs when there are fewer samples than the number of independent variables. In this work, we focus on the problem of stellar parameterization from low-mid resolution spectra, with blended absorption lines. We address this problem using an iterative algorithm to sequentially prune redundant features from synthetic PHOENIX spectra, and arrive at an optimal set of wavelengths with the strongest correlation with each of the output variables – T_{eff} , $\log g$, and $[\text{Fe}/\text{H}]$. We find that at any given resolution, most features (i.e., absorption lines) are not only redundant, but actually act as noise and decrease the accuracy of parameter retrieval.

1. Introduction

Technological developments in different domains have increased the amounts and complexity of data at an unprecedented pace, and astronomy is no exception to this trend. Availability of larger datasets may appear helpful for more effective decision making, but this is not so when this increase is primarily in data dimensionality. A large number of features can increase the noise of the data and thus the error of a learning algorithm. Feature selection is a solution for such problems. It reduces data dimensionality by removing irrelevant and redundant features. Besides maximizing model performance, other benefits include the ability to build simpler and faster models using only a subset of all features, as well as gaining a better understanding of the processes described by the data, by focusing on a selected subset of features.

In this paper, we deal with the issue of stellar characterization by spectral analysis, by employing a novel feature selection technique to automatically select absorption lines best suited for determination of the output parameters (T_{eff} , $\log g$, and $[\text{Fe}/\text{H}]$). Using synthetic PHOENIX spectra, we show that the proposed algorithm is able to improve parameter prediction accuracy, in addition to being able to robustly select the most important absorption lines in the wavelength range considered. This method is constructed in a modular fashion, and can be generalized to any regression task, within and outside of astronomy. At the time of writing this document, there exists only one astronomical publication dealing with feature selection (D’Isanto et al. 2018), and as such, we believe that the proposed method is an important addition to the literature.

2. Feature Selection Methods

Feature selection techniques can be divided into three categories, depending on how they interact with the predictor – ‘filter,’ ‘wrapper,’ and ‘embedded’ (see Guyon & Elisseeff 2003; Saeys et al. 2007, for reviews). Filters operate directly on the dataset and select subsets of features as a pre-processing step, independently of the chosen predictor. Wrappers, on the other hand, select a subset of features based on the output of the prediction model. Embedded methods work similarly to wrappers, but use internal information from the prediction model to do feature selection (Saeys et al. 2008). They often provide a good trade-off between performance and computational cost. More often than not, different feature selection algorithms will choose different feature subsets (Saeys et al. 2008). Ideally, we want any feature selection process not only to pick features with the best predictive power, but to also be robust – small changes in the input dataset, or different runs of the feature selection model, should not affect the selected features. Robustness of feature selection processes has received relatively little attention, and most work has rather focused on the stability of single-feature selection techniques (Křížek et al. 2007; Saeys et al. 2008; Raudys 2006). In this work, we explore the use of ensemble learning-based feature selection (Dietterich 2000) to yield a stable set of selected features (absorption lines), while also using an ensemble of regressors to better predict the output variables. To the best of our knowledge, the current work is the first such work.

3. Methodology

3.1. Data

We use absorption spectra from the synthetic PHOENIX library (Husser et al. 2013) to test our proposed algorithm. We select a total of 1800 spectra as follows: $T_{\text{eff}} = 3500$ to 7000 K in steps of 100 K, $\log g = 2.5$ to 5.0 dex in steps of 0.5 dex, and $[\text{Fe}/\text{H}] = -1$ to +1 dex in steps of 0.25 dex. The spectra were convolved from the native resolution of 500,000 down to 10000, and uniformly re-sampled onto a wavelength range of 5000 – 5450 Å. This was done to match the operating characteristics of MARVELS (Ge et al. 2008) (a low-medium resolution spectrograph commissioned as part of the Sloan Digital Sky Survey–III), since we plan to apply the proposed method to re-characterize stars observed with this instrument and compare the predicted parameter values to published ones. Finally, we picked 10% of all spectra (i.e., 180), distributed uniformly in the three-dimensional parameter space, as the training set, with the remaining 90% set aside as the test set.

3.2. Ensemble Feature Selection

Our goal is to formulate a feature selection strategy that not only reduces prediction error, but is also robust – both to variations in the training dataset, and to the initial conditions of the machine learning algorithms employed. Since features here refer to absorption lines, which have astrophysical origin and hence physical significance, it is imperative that any feature selection algorithm pick the same set of features over multiple runs. We achieve these goals using ‘ensembling’ (Dietterich 2000), i.e., combining different ML models to overcome their individual limitations. Specifically, we

use the techniques of ‘bagging’ (short for ‘bootstrapped aggregation’), and ‘stacking’ (combining the predictions, i.e., outputs, of one ML models as the input to another).

Using the training set of 180 stellar spectra, we create 1000 different ‘bags’ of data, each of which contains approximately 67 ($\sqrt{4500}$, see Friedman et al. (2001) for why we choose square root) features that are picked randomly (and with replacement across different ‘bags’) from the input 4500 absorption lines. For each of these ‘bags’, we create 100 bootstrapped versions (picking randomly and with replacement) of the training data (i.e., stellar spectra), and associate with them all a k-Nearest Neighbor regressor (Dudani 1976) with distance metric $|x-y|$. For each ‘bag’ we find the best k value by minimizing the ‘out-of-bag’ error as follows – we train the kNN on each of the 100 bootstrapped datasets, predict the output variable for stellar samples left out during bootstrapping, and average the rms error with respect to the ground truth values across all 100 datasets. We choose the value of k that minimizes this error as the final value for that ‘bag’. We rank the input features according to their average errors across all ‘bags’, discard the bottom 10%, and refit using three tree-based techniques: random forest, ada-boost, and extra trees regressor. We repeat this procedure of recursive backward elimination until the rms stops decreasing. The entire process is also repeated separately for each of the three parameters of interest.

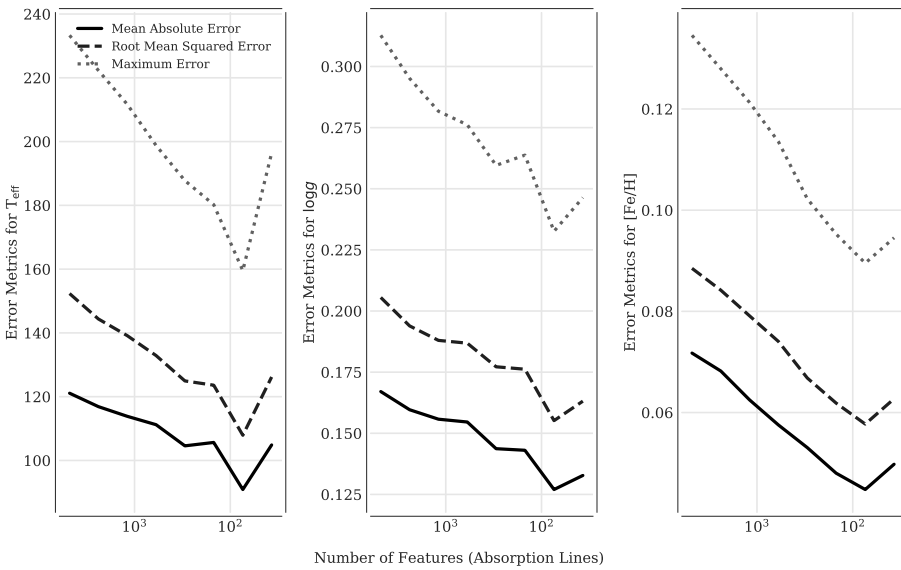


Figure 1. Error metrics as a function of number of features for T_{eff} , $\log g$, and $[\text{Fe}/\text{H}]$. We plot three different error metrics between ground-truth values and the respective ensemble-predicted values. As redundant features are removed, prediction errors decrease (moving from left to right). All metrics start increasing again after the inflection point, when the actually informative features start being trimmed. The x-coordinate of this inflection-point is the optimal number of features for predicting the respective parameters.

4. Results and Conclusions

We have proposed an ensemble-based feature selection algorithm for dealing with datasets with large number of features and small number of samples. We have demonstrated its efficacy by successfully characterizing synthetic PHOENIX spectra (with predicted variables being T_{eff} , $\log g$ and $[\text{Fe}/\text{H}]$) with 450 training samples and 4500 absorption lines (features) spanning the wavelength range from $\lambda = 5000 \text{ \AA}$ to 5450 \AA . We were able to successfully select approximately 100 absorption lines and improve prediction accuracy for all three variables at the same time. The proposed method uses an ensemble of k-Nearest Neighbors to robustly select features in a recursive backward elimination procedure, and another ensemble of predictors to actually predict the output variables. Figure 1 clearly illustrates the decrease in various error metrics as the redundant absorption lines (features) are eliminated using our proposed backward elimination method.

In the future, we plan to move to a probabilistic framework for feature selection, by using quantile regression as opposed to obtaining point estimates from our ensemble of predictors. This would naturally output an error range for all predicted variables rather than a single cross-validation error value. The method proposed here can, in principle, be applied to the calibration sample of any spectrograph to select absorption lines most suitable for parameter prediction; this would account for any instrument peculiarities in addition to capturing the relevant physics via the selected absorption lines. The features selected in this manner can then be used to parameterize any new observed star, provided it lies in the original parameter space. We also plan to explore the performance of the proposed feature selection method as a function of input signal-to-noise ratio.

References

- Dietterich, T. G. 2000, in International Workshop on Multiple Classifier Systems (Springer), 1
- D’Isanto, A., Cavuoti, S., Gieseke, F., & Polsterer, K. L. 2018, preprint arXiv:1803.10032
- Dudani, S. A. 1976, IEEE Transactions on Systems, Man, and Cybernetics, 325
- Friedman, J., Hastie, T., & Tibshirani, R. 2001, The Elements of Statistical Learning, vol. 1 (Springer Series in Statistics New York, NY, USA)
- Ge, J., Mahadevan, S., Lee, B., Wan, X., Zhao, B., van Eyken, J., Kane, S., Guo, P., Ford, E., Fleming, S., et al. 2008, in Extreme Solar Systems, vol. 398, 449
- Guyon, I., & Elisseeff, A. 2003, Journal of Machine Learning Research, 3, 1157
- Husser, T.-O., Wende-von Berg, S., Dreizler, S., Homeier, D., Reiners, A., Barman, T., & Hauschildt, P. H. 2013, Astronomy & Astrophysics, 553, A6
- Křížek, P., Kittler, J., & Hlaváč, V. 2007, in International Conference on Computer Analysis of Images and Patterns (Springer), 929
- Raudys, S. 2006, in Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR) (Springer), 622
- Saeys, Y., Abeel, T., & Van de Peer, Y. 2008, in Joint European Conference on Machine Learning and Knowledge Discovery in Databases (Springer), 313
- Saeys, Y., Inza, I., & Larrañaga, P. 2007, Bioinformatics, 23, 2507

A Method to Detect Radio Frequency Interference Based on Convolutional Neural Networks

C. Dai,¹ S.F. Zuo,^{2,3} W. Liu,¹ J.X. Li,² M. Zhu,¹ F.Q. Wu,² and X.C. Yu¹

¹*College of Information Science and Technology, Beijing Normal University, Beijing, Beijing, China; yuxianchuan@163.com*

²*National Astronomical Observatories, Chinese Academy of Sciences, Beijing, China*

³*University of Chinese Academy of Sciences, Beijing, China*

Abstract. RFI is an important challenge for radio astronomy. In this paper, we adopt a deep convolution neural network with a symmetrical structure, the U-Net, to detect RFI. The U-Net can perform the classification task of clean signal and RFI. It extracts the features of RFI for learning RFI distribution pattern and then calculates the probability value of RFI for each pixel. Then we set a threshold to get the results flagged by RFI. Experiments on Tianlai data (A radio telescope-array, the observing time is from 20:15:45 to 24:18:45 on 27th of September 2016, and the frequency is from 744MHz to 756MHz) show that, compared with the traditional RFI flagging method, this approach can get almost consistent results with satisfying accuracy and take into account the relationship between different baselines, which contributes to correctly and effectively flag RFI.

1. Introduction

Along with the rapid development of telecommunication, a variety of potential radio frequency interference (RFI) sources exist, varying in their application, frequency, waveform, and power (Lahtinen et al. 2017). RFI generated from diverse human-produced sources like electronic equipment, cell phones, GPS (Akeret et al. 2017) and so on can contaminate the weak radio band data. Therefore, studying methods of RFI detection and mitigation is imperative for weak signal extraction. The method in Offringa et al. (2010) called SumThreshold operates in the observed time-frequency data plane of a single baseline by first performing a smooth background fit and then detects RFI in the residual data by using a combinatorial threshold criterion. Although this method is simple and prevailing, it does not take into account the spatial information between data from different baselines. In this paper, we achieve the goal of RFI detection based on the U-Net (Ronneberger et al. 2015). Firstly, we adjust the structure of the U-Net and train it for feature extraction and distribution pattern, which are used to classify a clean signal and RFI. And then we get probability value of RFI for each pixel. Finally, we set a threshold to finish the classification task.

The paper is organized as follows. The structure of the U-Net is described in Section 2. Section 3 is devoted to introducing the process of RFI detection with the

U-Net and discussing the results by employing data from Tianlai. Section 4 is the conclusion.

2. Architecture of the U-Net

Most existing convolution neural networks (CNNs) perform classification tasks, where the output is the single classification label of an image. However, the U-Net enables classification of each pixel within the image, which is suitable and competitive for image segmentation. RFI detection can be regarded as a special task of image segmentation. As for RFI signals, they usually appear in the form of points, vertical or horizontal lines. Thus, we implement the U-Net of 14 layers with the Keras framework to detect RFI signals.

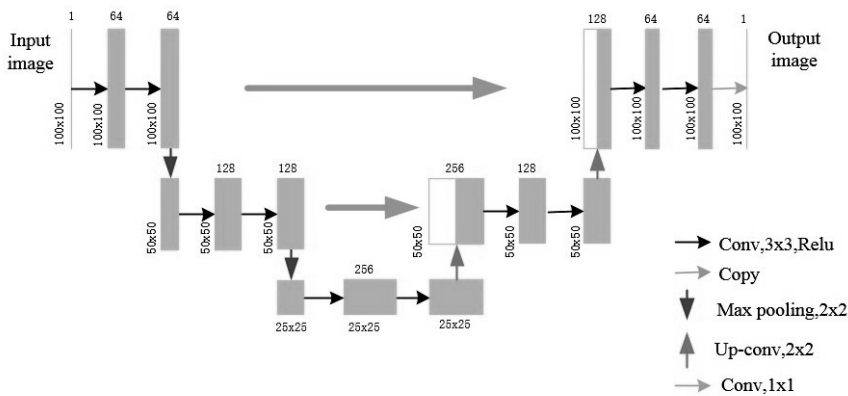


Figure 1. Architecture of the U-Net.

As demonstrated in Figure 1, the U-Net is a kind of extended CNN with symmetric architecture, which consists of a contracting path to capture context information and extract features and an expanding path to get precise localization. In the contracting process, after applying two convolutions on the input data, a down-sampling operation is operated. The early layers extract abstract and simple features and the deeper layers can learn complex features of the input data. And in the expanding path, as an up-sampling process, we concatenate the complex features extracted in each layer of the contracting process to higher layers.

3. Experiment

3.1. Preprocessing and model training

We use data observed by the Tianlai array (Chen 2012). The observing time is from 20:15:45 to 24:18:45 on 27th of September 2016. The frequency is from 744MHz to 756MHz and the number of baselines is 18528. We use the two-dimensional time-frequency observing data as input. For each baseline, the original time-frequency plane form is 3645 x 100 pixel. Before we apply the U-Net, we first remove the artificial

noise source signal in the data which is used for array calibration, so the number of time point is reduced to 3340. For the convenience of calculating, we transfer the form of one original time-frequency panel to thirty patches of 100 x 100 pixel. So we use 4800 patches as training data and 1600 patches as test data. We train our model with GeForce GTX 1080 Ti GPU.

3.2. RFI detection results

The SumThreshold Method in Offringa et al. (2010) flags RFI on a time-frequency 2D surface basing on different thresholds in different windows after 2D smoothing. Instead, the U-Net model can further learn more spatial information in the scale of the whole piece of data as well as data from different baselines. However, what we get from the U-Net are probability values of RFI for each pixel. And the results of the two methods are shown in Figure 2. Then we directly set a threshold of 0.5. If the probability value is greater than 0.5, the pixel is contaminated by RFI and flagged as 1. Otherwise, the pixel is flagged as 0. Also, compared to SumThreshold, we take data of baseline 171 as an example to show the flagging results in Figure 3 and the confusion matrix of the U-Net in Figure 4. And the precision rate, recall rate, and F1-score are recorded in Table 1.

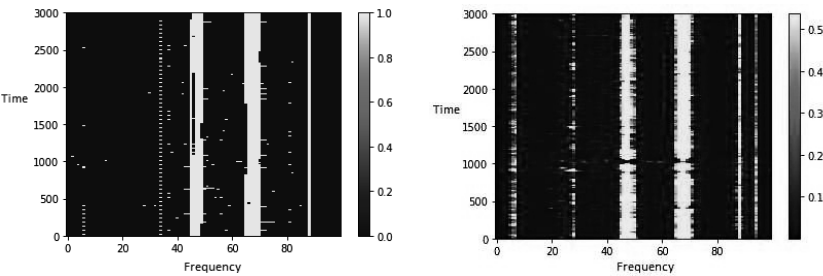


Figure 2. RFI flagging results of baseline 171 of SumThreshold and U-Net methods. Left: The RFI flagging result of SumThreshold. Right: The initial RFI flagging result of the U-Net.

Table 1. The precision and recall rate of RFI flagging of baseline 171.

Category	Precision rate	Recall rate	F1-score
False(Non-RFI)	0.97	0.98	0.98
True(RFI)	0.87	0.77	0.82

4. Conclusion

In this paper, in order to detect RFI signals effectively, we adopt a kind of prevailing deep convolution neural network, the U-Net. First, we preprocess the data to suit the

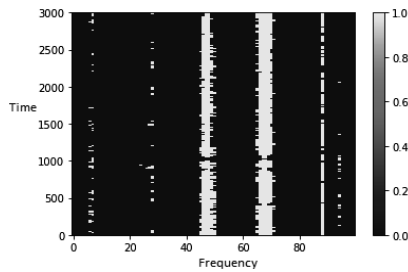


Figure 3. Improvement of the initial RFI flagging result of baseline 171 through setting a threshold of 0.5.

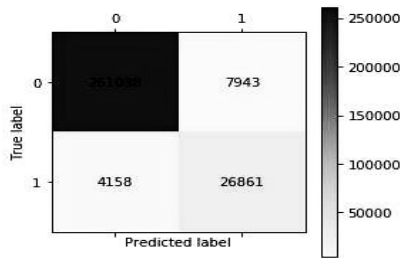


Figure 4. Confusion matrix result of baseline 171 of the U-Net

network, and then we train the U-Net with Adam(Adaptive Moment Estimation) to optimize the loss function. Finally, we detect RFI signals with the trained U-Net and obtain results with RFI flagging. We firstly use the U-Net on Tianlai data and obtain satisfying results. Meanwhile, we directly employ the method on real data rather than simulated data.

Acknowledgments. The study is supported by the National Natural Science Foundation of China under Grants No.11473044,41672323, Beijing Natural Science Foundation L172029, China Program of International S&T Cooperation 2016YFE0100300 and "the Interdisciplinary Research Funds of Beijing Normal University". Thanks for the data and help supported by the Cosmology Group of National Astronomical Observatories of China.

References

- Akeret, J., Chang, C., Lucchi, A., & Refregier, A. 2017, *Astronomy & Computing*, 18
- Chen, X. 2012, in *International Journal of Modern Physics Conference Series*, vol. 12 of International Journal of Modern Physics Conference Series, 256, 1212.6278
- Lahtinen, J., Uusitalo, J., Ruokokoski, T., & Ruoskanen, J. 2017, *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, PP, 1
- Offringa, A. R., De Bruyn, A. G., Biehl, M., Zaroubi, S., Bernardi, G., & Pandey, V. N. 2010, *Monthly Notices of the Royal Astronomical Society*, 405, 155
- Ronneberger, O., Fischer, P., & Brox, T. 2015, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234

Cherenkov Shower Detection Combining Probability Distributions from Convolutional Neural Networks

Mauricio Araya,^{1,2} Francisco Casas,² and Rodrigo Cáceres²

¹*Centro Científico Tecnológico de Valparaíso, Valparaíso, Chile;*
mauricio.araya@usm.cl

²*Departamento de Informática, UTFSM, Valparaíso, Chile*

Abstract. Ground-based gamma-ray observatories such as the Cherenkov Telescope Array presents new challenges for astronomical data analysis. The dynamics of the atmosphere and the complexity of Cherenkov shower are two uncertainty sources that needed to be embraced rather than corrected. As each telescope only has access to a separated patch, the partial information of each one has to be combined. For instance, when blots can be identified on the images, the application of Hillas parameters allows to identify the approximate direction to the projection's center. This information can be combined for several telescopes using stereoscopic reconstruction to converge on a single point. The limitation of this technique however is that it performs regressions to a predefined blot shapes, not using all the information contained in the images. Thus, deep learning techniques based on Convolutional Neural Networks have been applied with promising results. However, they rely on very large networks that process all the telescope images at once, which might not scale properly when dealing with large arrays. We propose to run several separate instances of an smaller network for each telescope, but that are able to retrieve a probability distribution instead of approximate coordinates for the sought point. This probability distribution can be arranged by the network so it can express certainty about the direction or the distance to the center of the projection separately. The distributions retrieved by all the telescopes can be combined to get a final probability distribution. Preliminary results shows the viability of this approach to identify the center and assign a confidence value to the result.

1. The Gamma-ray Reconstruction Problem

When a very-high energy gamma-ray photon hits the Earth's atmosphere, it produces a fast shower of particles, some of them traveling at ultra-relativistic speed, and therefore emitting Cherenkov light that can be detected by using Imaging Atmospheric Cherenkov Telescopes (IACTs) (Völk & Bernlöhr 2009). The analysis of the images produced by the telescopes, allow the reconstruction of the penetration depth in the atmosphere, its direction and energy. This will be done on a large scale and with high precision by the Cherenkov Telescope Array (CTA), to be installed in the next few years in Chile and Spain (Actis et al. 2011). The estimation of physical parameters from the images is a complex inverse problem that requires computing-intensive methods for simulations, inference and validation. The accuracy, performance and reliability of the algorithms involved in particle discrimination and gamma-ray reconstruction, are essential for science with IACTs, because the sensitivity and confidence on the whole observa-

tion depends on them. There are classical approaches for reconstructing gamma-rays from IACTs images, such as exploiting the known geometric properties of the showers (Hillas 1985), likelihood maximization of statistical models (De Naurois & Rolland 2009) or using fast monte-carlo simulations (Parsons & Hinton 2014). However, recent advances are strongly based on machine learning methods, such as random forests, boosted decision trees, and more recently, deep learning (Shilon et al. 2018; Mangano et al. 2018). The use of these data-driven techniques impose new challenges in terms of computational performance, correctness demonstration and uncertainty propagation.

2. Uncertain Multi-Observer Neural Network Assembly (UMONNA)

The general framework that we propose, called UMONNA, is formalized as follows. Let $T=\{t_1, t_2, ..., t_n\}$ be the set of images (and possibly additional data) received by a set of n observers, and y a value that wants to be predicted on the domain D from this images. The conventional machine learning approach would be to train the parameters W of a model f that retrieves an approximation \hat{y} of y from the set T , i.e., $\hat{y} = f(T|W)$. We propose a model that retrieves a Probability Density Distribution (PDF) for y on the domain D , based on the data of each observer t_i independently. For this, let M be a PDF parametrized by Σ_i that follows a model f :

$$\Sigma_i = f(t_i|W)$$

Then, the PDF for y is:

$$M(z|\Sigma_i) \quad \text{where } z \in D$$

This allows each observer i to retrieve wide probability distributions when their uncertainty is high, which may be the case when t_i doesn't contain enough information, this also allows them to express uncertainty in particular components of the domain D , e.g. distance but not direction. The PDFs retrieved from the observers are then merged within the domain D :

$$M^*(z) = \frac{\sqrt[n]{\prod_i M(z|\Sigma_i)}}{\int_D \sqrt[n]{\prod_i M(z|\Sigma_i)} dz}$$

The approximation \hat{y} of y can be computed from the distribution directly, for example by using the MAP criterion:

$$\hat{y} = \arg \max_{z \in D} M^*(z|\Sigma_i).$$

2.1. Proof of Concept: Simple UMONNA

As a proof of concept we generated a “dummy shower”: synthetic blots distributed with a clear pattern around a shower center and try to predict its position. We take spread patches of this canvas as the images that each telescope receives. We trained a Convolutional Neural Network (CNN) (Goodfellow et al. 2016) that retrieves a probability density distribution where each output of the M output neurons represents the probability of the center being at a certain angle (see Figure 1).

One of the main challenges of this idea is to choose the right loss function so that the retrieved PDFs can be evaluated correctly. When the PDF displays a high degree of certainty, this loss must be higher on errors and lower on right predictions than when it doesn't.

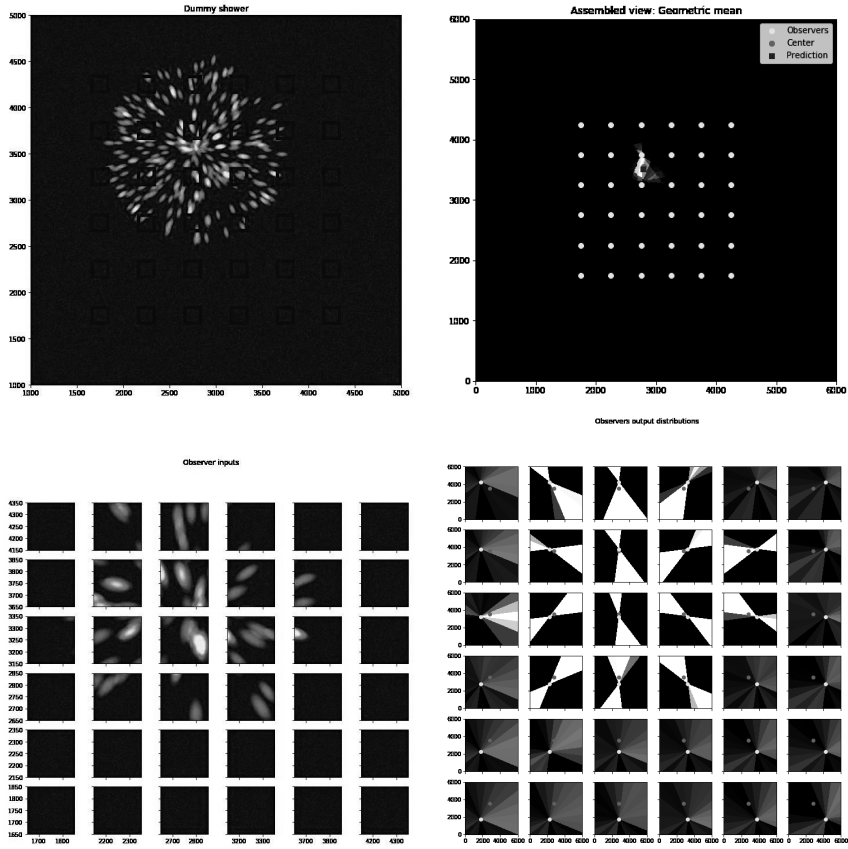


Figure 1. **Simple UMONNA.** The top-left image shows the synthetic shower over the domain D . The bottom-left image simulates the scattered view that the telescopes will observe. The bottom-right image presents the result of each network: the sought-point probability distribution based only on a single image reconstruction. At last, the top-right image shows the joint probability distribution and the maximum-a-posteriori (MAP) sought-point.

For this proof of concept, we selected a loss function that evaluates y in the output distribution to obtain an error value:

$$L(y, \Sigma_i) = \left(1 - \min \left\{ 1, \frac{M(y|\Sigma_i)}{C_{ap}} \right\} \right)^2$$

where C_{ap} is a limit for the PDF density to ensure that it is spread on other points of the domain D .

3. Conclusions and Future Work

The UMONNA approach separates a problem in smaller ones and provides a way to merge the results (i.e., divide and conquer strategy). This allows to work with simpler learning models, furthermore, by separating the observations, it increases the number training of samples by a factor of n . Another benefit of this approach is that it allows to measure the level of certainty of the assembly on different points of a continuous domain D . We believe that this could be used to detect simultaneous γ -ray phenomena occurrences through finding more than one local maximum in $M^*(.)$.

In terms of future work, we are beginning to train and test with simulated data using the Monte-Carlo simulators provided by CTA (Bernlöhr 2008). Also, we plan to estimate the energy and direction of the γ -rays, and not only the sought point as in this paper. On the model side, we would like to get smoother PDFs through interpolation, retrieve multivariate Gaussian distributions through deconvolutions, and include automatic derelativation for each telescope.

Acknowledgments. This work has been partially funded by CONICYT PIA/Basal FB0821, CONICYT - PFCHA/Magister Nacional/2018 - folio 22182114, and DGIIP PI-L-18-13.

References

- Actis, M., Agnetta, G., Aharonian, F., Akhperjanian, A., Aleksić, J., Aliu, E., Allan, D., Allekotte, I., Antico, F., Antonelli, L., et al. 2011, *Experimental Astronomy*, 32, 193
- Bernlöhr, K. 2008, *Astroparticle Physics*, 30, 149
- De Naurois, M., & Rolland, L. 2009, *Astroparticle Physics*, 32, 231
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. 2016, *Deep learning*, vol. 1 (MIT press Cambridge)
- Hillas, A. M. 1985
- Mangano, S., Delgado, C., Bernardos, M. I., Lallena, M., Vázquez, J. J. R., Consortium, C., et al. 2018, in *IAPR Workshop on Artificial Neural Networks in Pattern Recognition* (Springer), 243
- Parsons, R., & Hinton, J. 2014, *Astroparticle physics*, 56, 26
- Shilon, I., Kraus, M., Büchele, M., Egberts, K., Fischer, T., Holch, T. L., Lohse, T., Schwanke, U., Steppa, C., & Funk, S. 2018, arXiv preprint arXiv:1803.10698
- Völk, H. J., & Bernlöhr, K. 2009, *Experimental Astronomy*, 25, 173

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

A New Implementation of Deep Neural Network for Spatio-Spectral Analysis in X-Ray Astronomy

Hiroyoshi Iwasaki,¹ Yuto Ichinohe,¹ Yasunobu Uchiyama,¹ and
 Hiroya Yamaguchi²

¹*Department of Physics, Rikkyo University, 3-34-1 Nishi-Ikebukuro,
 Toshima-ku, Tokyo 171-8501, Japan; h.iwasaki@rikkyo.ac.jp*

²*ISAS, JAXA, 3-1-1 Yoshinodai, Chuo-ku, Sagamihara, Kanagawa 252-5210,
 Japan*

Abstract. Recent rapid developments in deep learning, which can implicitly capture structures in high-dimensional data, will lead to the opening of a new chapter of astronomical data analysis. As a new implementation of deep learning techniques in the fields of astronomy, we here report our application of Variational Auto-Encoder (VAE) using deep neural network to spatio-spectral analysis of the data from the *Chandra X-ray Observatory*, in the particular case of Tycho's supernova remnant. Previous applications of Machine Learning techniques to the analysis of SNRs have been limited to principal component analysis (Warren et al. 2005; Sato & Hughes 2017) and clustering without dimensional reduction (Burkey et al. 2013). We have implemented an unsupervised learning method combining VAE and Gaussian Mixture Model (GMM), where the reduction of dimensions of the observed data is performed by VAE and clustering in the feature space is by GMM. We have found that some characteristic features such as the iron knots in the southeastern region can be automatically recognized through this method. Our implementation exploits a new potential of deep learning in astronomical research.

1. Introduction

Machine learning (ML), especially deep learning has the potential to assist astronomical data analysis to take advantage of the rich information and extract the essential information without human bias from astronomical data, which are high-dimensional and complex.

Astronomical observations result in complex multi-dimensional data (spatial, temporal, and spectroscopic information). Thus conventional analyses can include human bias and oversights. In the plan of next X-ray observatory, the energy resolution dramatically increases (the spectral resolution of 2.5 eV up to 7 keV on the spatial resolution $\sim 5''$ with ~ 4000 pixels, *Athena*), which can be beyond human capacities. In such reasons, automatic and unbiased methods to discover features and analyses are expected.

In this work, we implemented a model to find characteristic spatial structures of a supernova remnant (SNR) from X-ray spectral information alone, and demonstrated the method on Tycho's SNR observed by *Chandra* which is one of the best benchmark objects.

2. Machine Learning Method

Unsupervised learning method suits the case to automatically find characteristic features because 1) labeling thousands or millions of data points highly costs, 2) in some case, it is unclear what to label even for human experts, and 3) hidden features may contain important physical meaning.

We explored the method combining two unsupervised learning methods, non-linear dimensional reduction using variational auto encoder (VAE; Kingma & Welling 2013) and clustering using Gaussian Mixture Model (GMM), for automatic investigation of spatial structures of a diffuse object for the first time. VAE is a deep neural network architecture, which can extract non-linear features from high-dimensional and big data well. We describe the details of VAE in the next section. GMM is a soft-clustering method which does not divide data distribution by hyperplanes but represents data with a superposition of some Gaussians. The probability that a data point belongs to a cluster is represented by the ratio of the value of the Gaussian corresponding to the cluster to the sum of the all Gaussian values at the data point, which is also referred to as responsibility.

2.1. Variational Auto Encoder

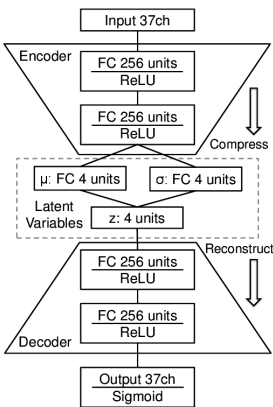


Figure 1. The diagram of VAE architecture. See the main text for the details of the network.

We used VAE summarized in Fig. 1 to reduce data dimension. VAE is an unsupervised deep learning architecture, and one of advanced models of the auto-encoder. Each box in Fig. 1 represents a layer of neurons. In VAE, a multidimensional Gaussian is assumed for the distribution of the latent variables. Unlike a normal auto-encoder whose encoder directly computes latent variables, the encoder of a VAE computes the means μ and variances σ . A set of latent variables \mathbf{z} is sampled from the multidimensional Gaussian represented by the means μ and the variances σ . The decoder reconstructs the original input from the set of the latent variables \mathbf{z} . The VAE trains to minimize the sum of the reconstruction error and the regularization error. The training was performed through 100 epochs with the batch size of 100.

2.2. Data Set of Chandra X-ray observation

The data set was treated as multi-color images of *Chandra* observations. Each pixel of the images was regarded as a 37 band spectrum. The bands were automatically chosen to have enough photons in each band in the spectrum of whole Tycho’s SNR. The bin size of the images was set to 3.94”, resulting in the image size of 146 × 143 pixels. The actual size of the training and the evaluation datasets were 150781 and 36808, respectively. All the observations in 2009 which has the longest total exposure were merged and used for the post-learning analysis.

3. Results

We reduced dimension of the data set using VAE and then performed the GMM clustering on the latent parameter μ . Fig. 2 compares the narrow-band flux image and the images of the GMM clustering. The unsupervised method automatically discovered the spatial structures, although no spatial information was used in the model. Some regions which appear similar in the RGB image in the left panel of Fig. 2, are classified to different categories using GMM in the middle panel of Fig. 2. This result shows the model can classify spectral features better than the RGB image.

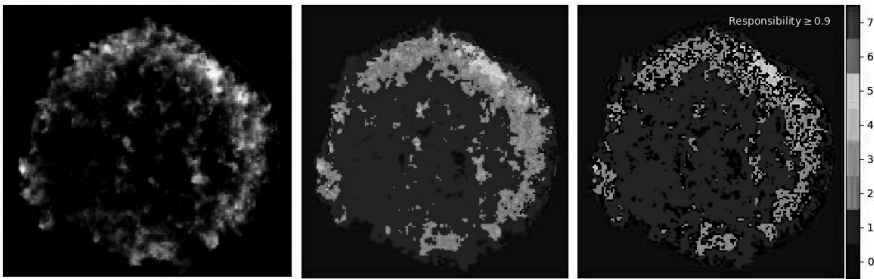


Figure 2. Tycho’s SNR observed by *Chandra* in 2009. *Left*: RGB image (Red: 0.7–0.95 keV Fe L, Green: 1.75–1.95 keV Si He α , Blue: 6.2–6.9 keV Fe K α). *Middle*: division of Tycho into GMM clusters. *Right*: selected pixels, which have responsibility above 90% are assigned the colors represented by the categories, and the other pixels are shown as black.

Table 1. GMM categories.

Category No.	location	feature
0	out of SNR	very dark, background
1	inner of SNR	dark region, swept ISM/CSM between FS and CD
2–4	rim of ejecta	bright ejecta
5	NW rim	weak Fe emitting ejecta
6	SE Fe knot	strong Fe emitting ejecta
7	shock, filaments	power-low radiation dominant

To investigate the spectral properties quantitatively, we fitted the spectrum representative of each category with the model of an absorbed power law for the continuum

emission plus Gaussians for the emission lines following Hayato et al. (2010). Based on the clustering, the spectra were extracted from all the pixels that are assigned to the certain category with the responsibilities above 90%, and combined them together.

The structures extracted by the method included physical features summarized in Table 1, which human experts of SNR had discovered as below. The category 0 is mainly the outside of Tycho's SNR, and contains some especially dark regions inside of the SNR. The category 1 is dark regions inside of the SNR, mainly including the unshocked ejecta and the swept interstellar medium (ISM) or circumstellar medium (CSM) between the forward shock (FS) and the contact discontinuity (CD). The category 2–4 are semi-circular region of bright ejecta. The category 5 is the ejecta emitting Fe lines weakly on the other hand emitting the lines of intermediate-mass elements (IME; i.e. Si, S, Ar, Ca) strongly at the north-west rim. As the result of the model fitting, we found the gradation of ionization state in the category 0–5, which is consistent of the gradation of the ionization time scale induced by the reverse shock (Sato & Hughes 2017). The category 6 located at the edge of the Fe knot in the south-east is ejecta having the strongest Fe $K\alpha$ line emission although the IME emissions are weaker. The Fe knot was analyzed by Yamaguchi et al. (2017) in detail. The category 7 spatially corresponds to the regions where the power-law emission is dominated such as the forward shock, the filament and stripe structure in the west of the SNR, and the arc inside of the south-east (Eriksen et al. 2011).

4. Conclusion

Our unsupervised ML method combined VAE and GMM automatically divided spatial structures which human experts of SNR had discovered in Tycho's SNR, one of the best-known SNRs. The demonstration showed the method can be a powerful tool for data analyses to exploit the rich information contained in SNR X-ray observations. The new method implemented in this work can be applied to other sources, such as galaxy clusters.

There is room for improvement in the deep learning architecture. An architecture of Wasserstein Auto-Encoder (WAE; Tolstikhin et al. 2017) or Gaussian Mixture VAE (GMMVAE; Dilokthanakul et al. 2016) may improve the latent manifold structure. A model using convolutional layers, e.g. convolutional VAE can be applicable to make use of the spatial information of data set.

References

- Burkey, M. T., Reynolds, S. P., Borkowski, K. J., & Blondin, J. M. 2013, *ApJ*, 764, 63
- Dilokthanakul, N., Mediano, P. A. M., Garnelo, M., et al. 2016, *ArXiv e-prints*. 1611.02648
- Eriksen, K. A., Hughes, J. P., Badenes, C., et al. 2011, *ApJ*, 728, L28
- Hayato, A., Yamaguchi, H., Tamagawa, T., et al. 2010, *ApJ*, 725, 894
- Kingma, D. P., & Welling, M. 2013, *ArXiv e-prints*. 1312.6114
- Sato, T., & Hughes, J. P. 2017, *ApJ*, 840, 112
- Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. 2017, *ArXiv e-prints*. 1711.01558
- Warren, J. S., Hughes, J. P., Badenes, C., et al. 2005, *ApJ*, 634, 376
- Yamaguchi, H., Hughes, J. P., Badenes, C., et al. 2017, *ApJ*, 834, 124

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Variable Star Classification using Multi-View Metric Learning

K.B. Johnston,¹ S.M. Caballero-Nieves,¹ A.M. Peter,² V. Petit,³ and R. Haber⁴

¹*Aerospace, Physics and Space Sciences Dept., Florida Institute of Technology,
150 W. University Blvd., Melbourne, FL, US; kyjohnst2000@my.fit.edu*

²*Computer Engineering and Sciences Dept., Florida Institute of Technology,
150 W. University Blvd., Melbourne, FL, US*

³*Physics and Astronomy Dept., University of Delaware, Newark, DE, USA*

⁴*Mathematical Sciences Dept., Florida Institute of Technology, 150 W.
University Blvd., Melbourne, FL, US*

Abstract. Comprehensive observations of variable stars can include time domain photometry in a multitude of filters, spectroscopy, estimates of color (e.g. U-B), etc. When it is considered that the time domain data can be further transformed via digital signal processing methodologies, the potential representations of the observed target star are limitless. Presented here is an initial review of multi-view classification as applied to variable star classification, to address this challenge.

1. Introduction

The classification of variable stars relies on a proper selection of features of interest and a classification framework that can support the linear separation of those features. Features should be selected that quantify the signature of the variability, i.e. its' structure and information content. Prior studies have generated a multitude of features (e.g., SSMM, Fourier Transform, Wavelet Transformation, DF, etc.) that attempt to completely differentiate or linearly separate various variable stars class types (Richards et al. 2012; Graham et al. 2013; Mahabal et al. 2017; Hinnert et al. 2018). How to process the complete set of features is an outstanding question.

Metric Learning has a number of benefits that are advantageous to the astronomer. First, metric learning uses k-NN classification to generate the decision space, k-NN provides instant clarity into the reasoning behind the classifiers decision (based on similarity, " x_i is closer to x_j than x_k "). Second, metric learning leverages side information (the supervised labels of the training data) to improve the metric, i.e. a transformation of the distance between points that favors the proposed goals: pull representatives from similar classes closer together and push representatives from different classes further apart, optimize to a low complexity metric via regularization, allow for feature dimensionality reduction, etc. (Bellet et al. 2015). Third, k-NN implemented as part of metric learning can be supported by other structures such as partitioning methods to allow for a rapid response time, despite a high number of training data (Faloutsos et al. 1994). Lastly, it can support the development of an anomaly detection functionality, which

has been shown to be necessary to generate meaningful data in astronomical datasets (Johnston & Peter 2017).

Multi-view learning can be leveraged to address the multitude of feature spaces or views that may be available to the astronomer for the purpose of classification. Multi-view learning can be roughly divided into three topic areas: 1) co-training, 2) multiple-kernel learning, and 3) subspace learning. This work will focus on the method of co-training, specifically metric co-training. The multi-view metric distance is defined as Equation 1:

$$d_M^2(x_i, x_j) = \sum_{k=1}^K w_k (x_i^k - x_j^k)^T \mathbf{M}_k (x_i^k - x_j^k) \quad (1)$$

where K is the number of views, x_i^k is the i^{th} observation and the k^{th} view for a given input. Presented here is a design that incorporates both metric learning and multi-view learning.

2. Theory and Design

Our proposal is an implementation of both the feature extraction and classifier for the purposes of multi-class identification, that can handle raw observed data. We implement two novel time domain feature space transforms, SSMM (Johnston & Peter 2017) and DF (Helfer et al. 2015), to demonstrate the utility of the metric learning and multi-view learning. It is not suggested that these features are going to be the best in all cases, nor are they the only choice as is apparent from Fulcher et al. (2013).

Large Margin Multi-Metric Learning (Hu et al. 2014, 2017) is an example of metric co-training; the designed objective function minimizes the objective function of the individual view, as well as the difference between view distances, simultaneously. The objective function for LM^3L is defined as Equation 2:

$$\min_{\mathbf{M}_1, \dots, \mathbf{M}_K} J = \sum_{k=1}^K w_k^p I_k + \lambda \sum_{k,l=1, k < l}^K \sum_{i,j} (d_{\mathbf{M}_k}^2(x_i^k, x_j^k) - d_{\mathbf{M}_l}^2(x_i^l, x_j^l))^2 \quad (2)$$

s.t. $\sum_{k=1}^K w_k = 1, w_k \geq 0, \lambda > 0$

where I_k is the objective function for a given k^{th} individual view (Equation 3):

$$\min_{\mathbf{M}_k} I_k = \sum_{i,j} h(\tau_k - y_{ij} (\mu_k - d_{\mathbf{M}_k}^2(x_i^k, x_j^k))) \quad (3)$$

where $h(x) = \max(x, 0)$ is the hinge loss function, $y_{ij} = 1$ when data are from the same class and $y_{ij} = -1$ otherwise, and τ_k and μ_k are threshold parameters that enforce the constraint $y_{ij} (\mu_k - d_{\mathbf{M}_k}^2(x_i^k, x_j^k)) > \tau_k$. In practice, optimizing \mathbf{M}_k requires enforcing the requirement $\mathbf{M}_k > 0$, which can be slow depending on the methodology used. Hu et al. (2014) transform the metric \mathbf{M}_k , following Weinberger et al. (2006), as $\mathbf{M} = \mathbf{L}^T \mathbf{L}$.

The algorithm operates as a two step process (alternating optimization) between the optimization of the decomposed metrics \mathbf{L}_k and the weighting between the views w_k . The iterative update to the \mathbf{L}_k estimate is generated via gradient for each view. Second, the metrics \mathbf{M}_k are fixed with the updated values and the individual weights

$w = [w_1, w_2, \dots, w_k]$ are estimated. The estimates for each weight can be given as Equation 4:

$$w_k = \frac{(1/I_k)^{1/(p-1)}}{\sum_{k=1}^K (1/I_k)^{1/(p-1)}} \quad (4)$$

These two steps are then repeated for each iteration until $|J^{(t)} - J^{(t-1)}| < \varepsilon$, i.e. some minimum is reached. The derivation of this algorithm is outlined in Hu et al. (2014), and the algorithm for optimization for LM^3L is given as their Algorithm 1..

2.1. Large Margin Multi-Metric Learning with Matrix Variates ($LM^3L - MV$)

Glanz & Carvalho (2013) define the matrix normal distribution as $X_i \sim MN(\mu, \Sigma_s, \Sigma_c)$, where X_i and μ are $p \times q$ matrices, Σ_s is a $p \times p$ matrix defining the row covariance, and Σ_c is a $q \times q$ matrix defining the column covariance. The Mahalanobis distance for the Matrix-Variate Multi-View case is given as Equation 5:

$$d_{\mathbf{U}_k, \mathbf{V}_k}(X_i^k, X_j^k) = \text{tr} \left[\mathbf{U}_k (X_i^k - X_j^k)^T \mathbf{V}_k (X_i^k - X_j^k) \right] \quad (5)$$

where \mathbf{U}_k and \mathbf{V}_k represents the covariance of the column and row respectively. The individual view objective function is constructed similar to the LMNN Weinberger et al. (2006) methodology; the joint, sub-view objective function is then Equation 6:

$$\begin{aligned} \min_{\mathbf{U}_k, \mathbf{V}_k} I_k = & \sum_{i,j} \eta_{ij}^k \cdot d_{\mathbf{U}_k, \mathbf{V}_k}(X_i^k, X_j^k) \\ & + \gamma \sum_{j \sim i, l} \eta_{ij}^k (1 - y_{il}) \cdot h \left[d_{\mathbf{U}_k, \mathbf{V}_k}(X_i^k, X_j^k) - d_{\mathbf{U}_k, \mathbf{V}_k}(X_i^k, X_l^k) + 1 \right] \\ & + \frac{\lambda}{2} \|\mathbf{U}_k\|_F^2 + \frac{\lambda}{2} \|\mathbf{V}_k\|_F^2 \end{aligned} \quad (6)$$

Similar to LM^3L the objective function is Equation 7:

$$\min_{\mathbf{U}_k, \mathbf{V}_k} J_k = w_k I_k + \mu \sum_{q=1, q \neq k}^K \sum_{i,j} \left(d_{\mathbf{U}_k, \mathbf{V}_k}(X_i^k, X_j^k) - d_{\mathbf{U}_i, \mathbf{V}_i}(X_i^q, X_j^q) \right)^2 \quad (7)$$

This objective design can be solved using gradient descent solver; to enforce the requirements of $\mathbf{U}_k > 0$ and $\mathbf{V}_k > 0$ we leverage the decomposition $\mathbf{U}_k = \mathbf{\Gamma}_k^T \mathbf{\Gamma}_k$ and $\mathbf{V}_k = \mathbf{N}_k^T \mathbf{N}_k$ and find the gradient of the objective function with respect to the decomposed matrices $\mathbf{\Gamma}_k$ and \mathbf{N}_k . Weights per view can be estimated using the same procedure as in LM^3L . The implementation of distance in the multi-view case, i.e. implementation of distance used in the k-NN algorithm is just the weighted average of Equation 5 over all views.

3. Conclusion

Optimal parameters are found for the LM^3L algorithm LINEAR data (following a standard 5-fold cross-validation procedure). The trained classifier is applied to the test data, the confusion matrices resulting from the application to LINEAR data are presented as an example in Table 1:

The classification of variable stars relies on a proper selection of features of interest and a classification framework that can support the linear separation of those features. Features should be selected that quantify the signature of the variability, i.e. its'

Table 1. LINEAR Confusion Matrix via LM^3L

Error Rate	RL (ab)	δ S / SP	AI	RL (c)	CB	Miss
RR Lyr (ab)	0.9927	0	0	0.00648	0.0009	0
δ Scu / SX Phe	0.0370	0.9259	0	0	0	0.037
Algol	0.0073	0	0.7737	0	0.218	0
RR Lyr (c)	0.0485	0	0.0027	0.9434	0.0054	0
Contact Binary	0.0034	0	0.0377	0.0011	0.9577	0

structure and information content. To support the set of high-dimensionality features, or views, multi-view metric learning is investigated as a viable design. Multi-view learning provides an avenue for integrating multiple transforms to generate a superior classifier. Future research will include methods for addressing high dimensionality matrix data (e.g. SSMM), applying the designed classifier (LM^3L) to the datasets, improving the parallelization of the design presented, and implementing community standard workarounds for large dataset data (i.e., on-line learning, stochastic/batch gradient descent methods, k-d tree... etc.).

References

Bellet, A., Habrard, A., & Sebban, M. 2015, Synthesis Lectures on Artificial Intelligence and Machine Learning, 9, 1

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. 1994, Fast subsequence matching in time-series databases, vol. 23 (ACM)

Fulcher, B. D., Little, M. A., & Jones, N. S. 2013, Journal of the Royal Society Interface, 10, 20130048

Glanz, H., & Carvalho, L. 2013, arXiv preprint arXiv:1309.6609

Graham, M. J., Djorgovski, S., Mahabal, A. A., Donalek, C., & Drake, A. J. 2013, Monthly Notices of the Royal Astronomical Society, 431, 2371

Helfer, E., Smith, B., Haber, R., & A, P. 2015, Statistical Analysis of Functional Data, Tech. rep., Florida Institute of Technology

Hinners, T. A., Tat, K., & Thorp, R. 2018, The Astronomical Journal, 156, 7

Hu, J., Lu, J., Tan, Y. P., Yuan, J., & Zhou, J. 2017, IEEE Transactions on Circuits and Systems for Video Technology, PP, 1

Hu, J., Lu, J., Yuan, J., & Tan, Y.-P. 2014, in Asian Conference on Computer Vision (Springer), 252

Johnston, K. B., & Peter, A. M. 2017, New Astronomy, 50, 1

Mahabal, A., Sheth, K., Gieseke, F., Pai, A., Djorgovski, S. G., Drake, A., & Graham, M. 2017, in Computational Intelligence (SSCI), 2017 IEEE Symposium Series on (IEEE), 1

Richards, J. W., Starr, D. L., Miller, A. A., Bloom, J. S., Butler, N. R., Brink, H., & Crellin-Quick, A. 2012, The Astrophysical Journal Supplement Series, 203, 32

Weinberger, K. Q., Blitzer, J., & Saul, L. K. 2006, in Advances in neural information processing systems, 1473

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Multiscale Spatial Analysis of Young Stars Complex using the dbscan Clustering Algorithm

Isabelle Joncour,^{1,2} Estelle Moraux,¹ Gaspard Duchêne,^{1,3} and Lee Mundy²

¹*UGA, Grenoble, France; isabelle.joncour@univ-grenoble-alpes.fr*

²*UMD, College Park, MD, USA*

³*UC Berkeley, CA, USA*

Abstract. Clustering and spatial substructures studies of young stellar objects (YSOs) in star forming regions are key tracers to identify (1) the properties of the birth sites and (2) the dynamical evolution cluster structure with time. This work presents an approach identifying the multilevel topological substructures of star forming regions. In this preliminary work, we use recursively the dbscan density based clustering algorithm to identify the spatial multilevel substructures. From these multilevel substructures we build a clusterTree, the analogous of dendrogram, and an associated multilevel spectrum, and we show that they can be used to characterize the YSO spatial distribution. We first apply this procedure on different types of distributions (uniform, fractal, Plummer). We then apply this procedure to analyze the Taurus YSOs complex.

1. Introduction

The young stars form from dense molecular cloud material (Lada & Lada 2003). The fragmentation and the gravitational collapse of the densest parts of the cloud lead to the formation of young binaries and multiple stars as well stellar groups or larger stellar clusters. Thus, studying the clustering of young stars provides crucial information about the cloud structures and processes that produce stars. Later, the dynamical interactions between stars as well as the removal of interstellar gas may lead to the dispersion of loose stellar groups. In that respect, studying the evolution of the grouping pattern gives us information on the dynamical evolution of YSOs as go from embedded very young objects to revealed T Tauri stars. That's why identifying reliable spatial patterns and substructures in star forming regions is important and we need to use reliable tools to extract them.

Among machine learning techniques, unsupervised clustering is the process of grouping objects with similar properties together without assuming prior knowledge of the type or number of clusters. It's a long standing area of studies and research work; the most common ones are the k-means clustering types of partition algorithms and the hierarchical clustering such as single linkage (see for a review Appendix C in Joncour et al. (2018)). We tested a number of different algorithms and came to the conclusion that the density-based ones are the best for our objective of detecting spatial substructures, i.e., clusters defined as spatial regions of higher density separated by local drops in the surrounding density. We identify the dbscan algorithm as a powerful tool to par-

tition the data in clusters based on the density-based connectivity criteria (Ester et al. 1996) and we use it to identify the densest parts of the Taurus YSOs complex located at 140pc from the Sun.

2. Method

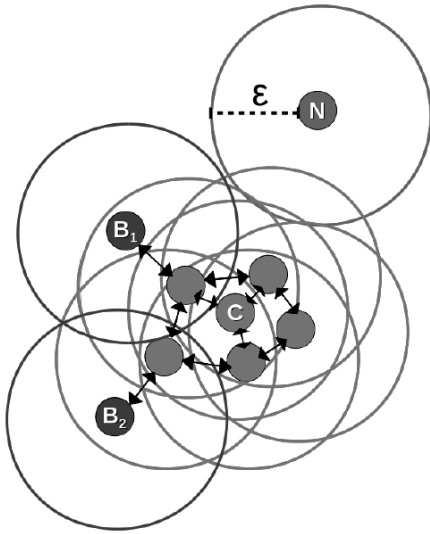


Figure 1. dbSCAN algorithm applied to a 9 points data set with $N_{min} = 4$ and radius ϵ . A C core cluster point (green) contains a minimum of 4 points (including the point itself) within a radius of ϵ . B_1 and B_2 points are density-reachable from C via core points so are border points (purple) of the cluster, whilst point N is not density reachable from core points so is designed as Noise (red).

Given a set of points in space, the dbSCAN algorithm groups together points provided that: (1) they are close neighbors (separated by a distance less than ϵ) and (2) have in common the same minimum local density, i.e., they have at least the same minimum of points N_{min} within a sphere of radius ϵ around each point. If this last condition is fulfilled for the two points, the points are said to be core-points and directly density-reachable. Two points in a cluster are said to be simply density-reachable if there is a path between these 2 points where each point along the path is directly reachable from the previous point. All the points along the path are then core-points, except the two extreme points at the ends of the path for which the condition on the minimal local density may not be fulfilled; they are then called border points. A point that is not reachable from any other point is called a noise point or an outlier. A cluster is formed by all the points that are reachable from all the other points (core or border points), and contain at least one core point (see Figure 1).

The main strengths of the dbSCAN algorithm (see Appendix B in Joncour et al. (2018)) are: (1) it is able to handle noise and outlier points to avoid spurious chaining effect, (2) the computational complexity time grows as N

$\log(N)$ when using a KD-tree and (3) the two free parameters N_{min} and ϵ can be optimized to the scientific goal.

In our last work (Joncour et al. 2018), we set the value of these two parameters based on a nearest neighbor statistics analysis to detect, with a high level of confidence (99.85%), 20 overdense structures in the YSOs Taurus complex. We show notably that these structures, which we called NESTs (for Nested Elementary Structures), are the preferred sites of stellar birth. These NESTs exhibit an evolutionary status based on the proportion of the Class of the YSOs they shelter, from the most embedded objects (Class 0) up to the more evolved Class III YSOs that have mostly cleared out the cir-

cumstellar gas. The question we raise now in this work is whether we can perform a multi-scale analysis of the spatial distribution to complement the one-level analysis and thereby obtain the full range of structures. In this preliminary work, we use recursively the dbscan algorithm keeping the N_{min} parameter at the same value while sweeping the full range of ϵ , from the scale of the whole region down to wide pair separations. For a given spatial star distribution, we identify at each spatial scale (value of ϵ) the density-connected stars. Based on that analysis, we derive (1) the density components spectrum defined as the distribution of detected density-connected star groups as a function of the spatial scale and (2) the clusterTree object (dendrogram analogue) associated to the full multilevel topology of the density components. The ClusterTree is built from an optimal, but still arbitrary, ordering of the stars based on their inter-distances (Y-axis), each node being a connected component at the ϵ -level (X-axis). The ClusterTree branches represent the inclusion of the connected components from one level to next other. To quantify the complexity of the star forming region spatial structure, we then further introduce the Strahler order as an indicator of the level and depth of the connected components (Strahler 1957). The leaves of the ClusterTree start with a Strahler order value of unity; when two or more substructures merge at a given level, the resulting substructure keeps the value of the higher Strahler order of the two merged components, or is increased by one if the Strahler order of the two components are equal. We use this framework to analyze (1) three example density profile types: fractal, Plummer, and random distributions in Figure 2 and (2) the Taurus star forming complex in Figure 3. The example density distributions were generated by the McLuster Fortran program (Küpper et al. 2011); the Taurus data are from Joncour et al. (2018).

3. Conclusion

We have set up a methodology and new tools to perform multiscale studies of spatial distributions of young stars using recursively the dbscan algorithm. These tools can characterize and quantify hierarchical structure in the presence of “noise,” non-clustered points. As the next step, we are implementing a methodology for expressing the level confidence of components at each level using nearest neighbor statistics criteria (Joncour et al., in prep).

References

- Ester, M., Kriegel, H., Sander, J., & Xu, X. 1996, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA, 226. URL <http://www.aaai.org/Library/KDD/1996/kdd96-037.php>
- Joncour, I., Duchêne, G., Moraux and, E., & Motte, F. 2018, ArXiv e-prints. 1809.02380
- Küpper, A. H. W., Maschberger, T., Kroupa, P., & Baumgardt, H. 2011, MNRAS, 417, 2300. 1107.2395
- Lada, C. J., & Lada, E. A. 2003, ARA&A, 41, 57. astro-ph/0301540
- Strahler, A. N. 1957, American Geophysical Union Transactions, 38(6), 912

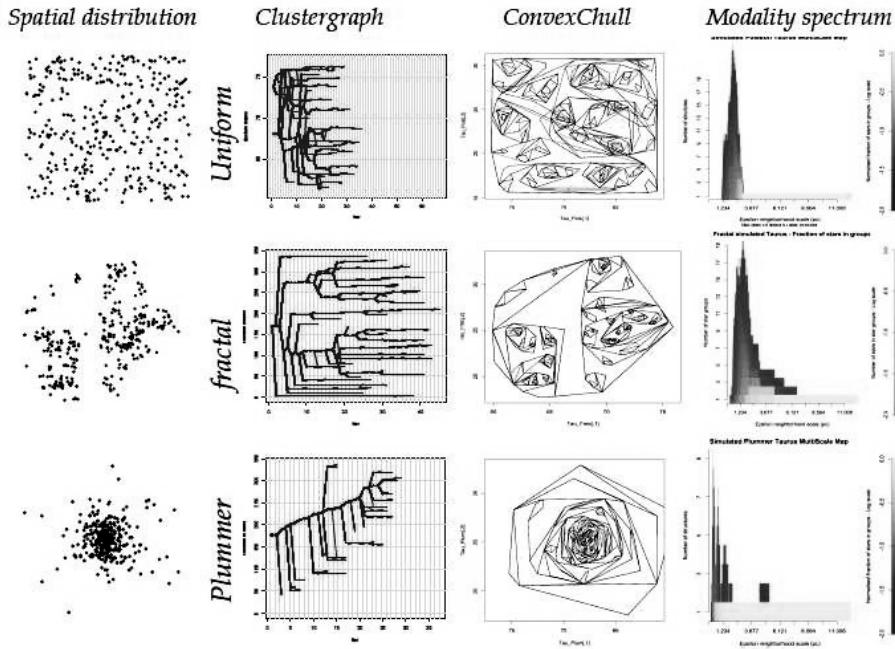


Figure 2. Multiscale analysis of the 3 example star distributions. From left to right, the spatial distribution, the ClusterTree, the multiscale connected components seen as convex hulls and finally the color-coded component spectrum (labeled multiscale spectrum), with the ϵ -radius on X-axis and the number of detected connected components on the Y-axis, from the yellow richer (in terms of number of stars contained within the component) components to light purple poorer components.

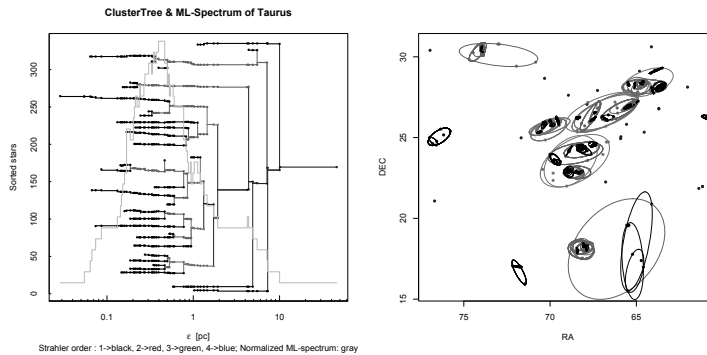


Figure 3. Left: the ClusterTree of Taurus overlaid on the multiscale spectrum in gray. The branches are color coded based on the Strahler order value from 1 to 4 (resp. black, red, green and blue). Right: the multiscale components as obtained by a spanning ellipsoid fit. They are color code based on the 1 to 3 Strahler order of the components. The stars within the 4 Strahler order components are left in blue.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

LAMOST DR5 Spectral Clustering for Stellar Templates Construction

Xiao Kong and A-Li Luo

National Astronomical Observatories, Chinese Academy of Sciences, Beijing, China; kongx@nao.cas.cn

Abstract. The large sky area multi-object fiber spectroscopic telescope (LAMOST) released the fifth spectral data (DR5) this year, containing 8,171,443 stars; 153,090 galaxies; 51,133 quasi-stellar objects (QSO); and 642,178 unknown types, all of which are classified by the LAMOST 1D pipeline. This pipeline is used for spectral analysis, aiming to determine the spectral type and redshifts of the spectra observed by LAMOST by matching them with spectral templates. Using template matching, we divide all the DR5 spectra into various groups according to their spectral type and signal-to-noise ratio. Then, we adopt k-means to build 500 cluster centers of spectra. After visual inspection, we select 197 centers that can be used as stellar templates for the pipeline. These templates are supposed to increase the number of types and the accuracy of the classification. In the end 19 cluster centers remain to be identified.

1. Introduction

The large sky area multi-object fiber spectroscopic telescope (LAMOST, Cui et al. (2012)), which can capture 4,000 objects during one exposure, is a special quasi-meridian reflecting Schmidt telescope located in Xinglong Station of National Astronomical Observatory, China (Yao et al. 2012). It has begun to release spectral data since 2012 (prior data release, PDR). At the beginning of 2018, LAMOST released its 5th spectral data, DR5, including 9,017,844 spectra (Table 1).

Table 1. Number of released spectra from LAMOST DR of each year.

	DR1	DR2	DR3	DR4	DR5
Star	1,944,329	3,784,461	5,268,687	6,856,896	8,171,443
Galaxy	12,082	37,206	61,815	118,657	153,090
QSO	5,017	8,630	16,351	36,374	51,133
Unknown	243,268	306,185	408,273	652,146	642,178
Total	2,204,696	4,136,482	5,755,126	7,664,073	9,017,844

All the spectral types are assigned by LAMOST 1D pipeline using template matching. The current stellar templates (Wei et al. 2014) were constructed from spectral data of LAMOST DR1 (Luo et al. 2015). As shown in Table 2, there are 183 stellar spectra that served as templates for classification, and they can help pipeline software to classify the stellar spectra at a high correction ratio.

Table 2. Number of spectra within the current LAMOST stellar template.

Subclass	O	B	A	F	G	K	M	Carbon	CV	DoubleStar	WD	Total
Number	2	2	49	25	24	36	38	3	1	1	2	183

With the number of stellar spectra photoed by the LAMOST increasing at a significant rate, however, some more rare objects appear in the vast amount of spectral data sets that cannot be identified by the pipeline using the current templates, e.g., DB white dwarfs. We utilize all the ≈ 9 million spectra from LAMOST DR5 as the material of new templates for LAMOST 1D pipeline.

2. Clustering

The basic idea of construction is hierarchical clustering: (1) dividing all the spectra into different groups according to their spectral type; (2) clustering spectra within each group into different cluster centers; (3) re-clustering all the centers.

In the beginning, full-spectra template-matching is adopted to assign each spectrum to a specific type (could be any type from the kinds listed in Table 2, QSO, and galaxy) and then the redshift (z) can be obtained. Considering the effect of noise, the spectra from each type are divided into two groups: signal-to-noise ratio ($S/N \leq$ and > 10). Afterward, 24 groups are built.

We then shift all spectra to the rest frame and set 100 cluster centers for each group before employing k-means (MacQueen 1967) to these groups respectively. All the 2,400 cluster centers adopted k-means again, and 500 centers are screened. Different centers correspond to various number of spectra.

Centers corresponding to spectra less than 100 are abandoned at first. We visually inspect all of them and discard 181 centers due to low spectral quality or the similarity between different group centers. Furthermore, the types of the remaining clustering centers are assigned by the subclass of the spectra from LAMOST DR5. In total, 197 centers could be served as the stellar templates, and their spectral types are illustrated in Table 3. In this table, if more than 80% of the spectra corresponding to one cluster center is of a specific type, then this center would be assigned to this type.

Meanwhile, 19 templates whose subclasses are difficult to determine are needed to be identified in our further work.

Some of the cluster centers are illustrated in the figures below.

References

Cui, X.-Q., Zhao, Y.-H., Chu, Y.-Q., & et al. 2012, Research in Astronomy and Astrophysics, 12, 1197
Luo, A.-L., Zhao, Y.-H., Zhao, G., & et al. 2015, Research in Astronomy and Astrophysics, 15, 1095
MacQueen, J. 1967, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics (Berkeley, Calif.: University of California Press), 281. URL <https://projecteuclid.org/euclid.bsmsp/1200512992>
Wei, P., Luo, A., Li, Y., & et al. 2014, AJ, 147, 101

Table 3. The main information of the cluster centers with certain spectral types. Each spectral type in the column one may include more detailed classification, e.g., A refers to A0, A1, A2 ... A9.

Subclass	Number
O	4,942
B	9,834
A	399,412
F	1,914,286
G	3,045,235
K	1,034,761
M	342,624
WD	7,391
Carbon	5,348
CV	1,581
DoubleStar	3,054

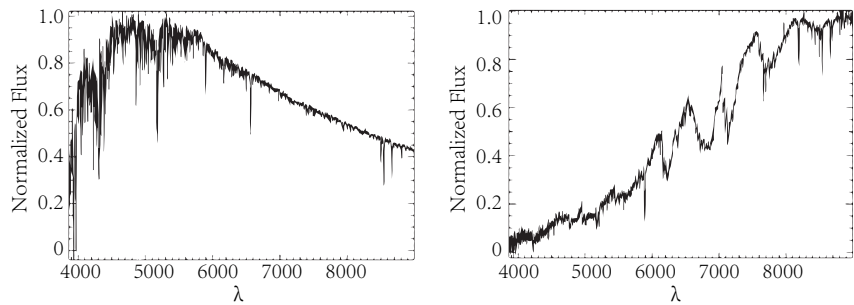


Figure 1. Two cluster centers that are assigned to certain types. *left:* star, G5. *right:* star, M4.

Yao, S., Liu, C., Zhang, H.-T., & et al. 2012, Research in Astronomy and Astrophysics, 12, 772. 1206.3574

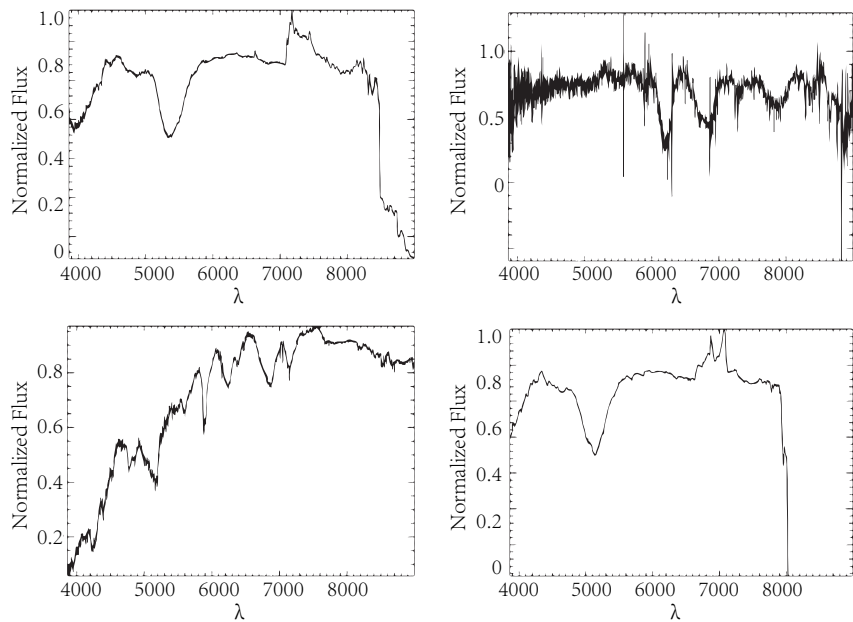


Figure 2. Some cluster centers that need to be identified in the next step.



Break time (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Saving Endangered Animals with Astro-Ecology

Paul Ross McWhirter, Joshua Veitch-Michaelis, Claire Burke, Marco C. Lam,
and Steven N. Longmore

*Astrophysics Research Institute, Liverpool John Moores University, Liverpool,
UK; P.R.McWhirter@ljmu.ac.uk*

Abstract. Conservation science is experiencing an unprecedented challenge in identifying and protecting endangered species across the world. The large stretches of land and sea require innovative solutions for the monitoring of endangered populations. Drones equipped with high resolution cameras with supporting data from satellites have helped to mitigate these challenges. Unfortunately, it is difficult to detect animals from optical images when they might only be a matter of a few pixels across.

By deploying thermal infrared cameras on drones to detect animals from their body heat, they can be detected despite their small size in the images. In the thermal infrared band, animals appear as bright sources on a dark, colder background. Through the use of astronomical source detection techniques, these bright animals can be detected although other warm objects lead to false detections. In this paper we demonstrate a technique which uses modern computer vision to build on astronomical source detection algorithms to create a model for the detection and classification of animal thermal profiles in the presence of other warm objects. Using a dataset from Chester Zoo in the UK, we trained a model using 972 frames from a video of the chimpanzee enclosure and achieved excellent results with a training loss of 0.81 and minimal false detections of warm environmental sources.

1. Introduction

Drones equipped with thermal cameras offer an excellent method of studying populations of animals. With the increasing volume of thermal data collected by our drones, the initial manual classifications of animal populations became unfeasible given the quantity of the collected thermal images. The nature of these images attracted astrophysicists to join the project employing the photutils package (Longmore et al. 2017). The thermal data was thresholded to reveal the brightest pixels in the images as proposed objects which are then analysed to determine if the warm pixels have a distinct border. This method performed well on data with a cool thermal background but this could change strongly depending on the time of day and the local climate. Using the Moderate Resolution Imaging Spectroradiometer satellites (MODIS), a model of land surface temperature variation across daily and yearly timescales was produced (Burke et al. 2018). This allowed for future data gathering flights to be optimised for the local conditions prior to the expedition. The atmosphere also results in an absorption effect on the observed sources in the thermal infrared primarily due to water content as a function of the distance from the camera to the source and the air temperature. The variability of the thermal data requires the development of a preprocessing pipeline to

calibrate the thermal images such that the thermal background has a zero to minimal contribution to the pixel intensities allowing the warmer objects to be identified.

We are developing a machine learning detection and classification system utilising the approaches in modern computer vision. This task requires a large set of labelled training data of multiple species. Whilst we have collected a large quantity of data, it would take a long time to manually identify the sources of interest. To address this problem we are using citizen science through Zooniverse. The Zooniverse site requests users place bounding boxes around the different species of animals seen in the thermal data and classify the identified animals. Upon the completion of the citizen science workloads, the training data can then be applied to a selection of computer vision machine learning algorithms to produce models for the detection and classification of animals from their thermal profiles. While the Zooniverse project progresses through the catalog of thermal data we have collected, an initial set of machine learning models have been trained using a combination of carefully selected data processed by a thresholding and contour detection pipeline and manually labelled images.

The rest of this paper is laid out as follows. In §2 state-of-the-art, machine learned, computer vision methods are discussed for detection and classification. Then, in §3, the results of a proof-of-concept model are demonstrated and the next directions to action these methods into a fully realised pipeline are suggested.

2. Detection and Classification

Computer vision is an important field involved in the processing of single and multi-colour image-based data using suites of algorithms designed to recognise features such as shapes, edges and corners. During the early 2010s machine learning and computer vision were united in an impressive way through the development of Deep Convolutional Neural Networks (CNNs) (Krizhevsky et al. 2012). Convolution layers share the same weights for every combination of pixels on the image of size equal to small, connected regions, introducing translation invariance allowing features to be learned regardless of their position in the images. It is this sharing of parameters which grants the layer its name as this is equivalent to the convolution of the input pixels and the weighting parameters.

CNNs are clearly well suited to providing classifications of images of objects but they natively do not detect multiple objects in an image or provide multiple classifications if the image contains multiple objects of interest. As multiple object detection is a substantial component of the desired data analysis pipeline, these methods must be augmented. Using CNNs for object detection was initially proposed using sliding windows of multiple scales to initially perform a binary classification, does the cropped image in this window contain an object of interest or just background (Girshick 2016). The windows which pass a given object probability threshold are then further classified by a multiclass CNN model into class probabilities.

This method of object recognition, the combination of detection and classification, has achieved good performance. Unfortunately, due to the multiple classifications per image or video frame, they struggle to operate in realtime. Redmon et al. (2016) suggested a more computationally efficient approach using a topology named YOLO: You Only Look Once. As the name suggests, instead of having the convolutional network make repeated classifications for every sliding windowed cropped image, the full images are input into their network which then splits the images into a grid. Any grid cell

which contains the center point of an object become responsible for drawing the bounding box around the object. These grid cells can define a number of bounding boxes on various scales which are optimised using regression into an optimal position for each detected object. The output contains the thresholded probability that each grid cell contains the center of a detected object, the optimal bounding box coordinates for the detected object for each grid cell and the classification vector showing the confidence in the classification of the detected objects.

In the subsequent three years, this method has been improved into YOLOv3 introducing improvements specifically suited to the analysis of thermal drone images. There is improved performance at identifying and positioning good bounding boxes around multiple smaller objects with poorer results with large objects. Objects large enough to cause this issue in the drone data are unlikely. Interestingly this is a complete reversal of the performance of the original method.

3. Results and Conclusion

The demonstration model was trained using thermal infrared data collected by a FLIR Tau 640 thermal imaging camera. The model was trained from thermal data collected during the morning of 30th of January, 2018. The thermal camera was mounted on a stationary mount viewing the chimpanzee enclosure at Chester Zoo. The Chester Zoo Chimpanzee data consists of 20 videos with 19435 frames. The temperature scale was adjusted on a by-video basis to minimize the thermal background. The colour adjustment is set to greyscale as the thermal camera does not collect colour information and simply detects all photons over the thermal infrared passband. To create the training data, every 20th frame of each of the 20 videos was extracted and exported as an image file. This training uses approximately 5% of the total frames.

Each of the training images required all objects of interest to be located and recorded. As the model was trained using the YOLOv3 algorithm, a piece of software named YOLO-mark was used to provide bounding box coordinates for objects of interesting classes using a user-friendly interface. The training frames consisted of a view over the Chimpanzee enclosure at a shallow angle. This results in the size of interesting objects varying based on their distance to the thermal camera. This can be problematic as CNNs are not natively scale-invariant and therefore the training data requires sufficient examples of each of the object classes across all the size scales. Three dominant classes of animal were identified from the 972 training frames. Firstly there are a variety of birds, consisting of both small flying birds and ducks within the nearby pond and on the land looking for food. The second class is human consisting of a number of people walking along a path in the top corner of the frames sometimes with bicycles and, for a specific time region, zookeepers moving about the enclosure places food for the chimpanzees. The chimpanzees are the final class and are present in a great number of the training frames both on the ground and climbing on trees and supports.

Over two days the 972 training images were manually inspected and bounding boxes placed using YOLO-mark. This process is greatly time consuming considering 972 frames is not a substantial training set for deep networks. Ideally use of some combination of an automated, but likely more inaccurate, object detection method or the Zooniverse citizen science can be used to greatly improve the efficiency of producing viable training datasets. When this process of manual object detection on the training data is complete, each frame, saved as a jpeg image, is accompanied by a text file

containing 5 columns and n rows where n is the number of detected objects in the frame. The 5 columns represent integer numbers where the first column indicates the class of an object detected in the frame, and the next four columns contain coordinates for the bounding box containing the detected object.

An Nvidia Geforce GTX 1080 Ti graphics card was used to train the models. The model was trained over 96 hours with 80,000 iterations where each iteration is a set of 64 randomly selected training images processed in 8 batches of 8 images. The final model has therefore been trained over $80,000 \times 64 = 5,120,000$ images. The training epochs measurement is defined as the number of complete passes over the training data and is calculated by dividing the total number of trained images by the number of images in the training set. Using this calculation, the first model at Chester Zoo was trained for 5,267 epochs. The final training loss as calculated by the YOLOv3 loss function is 0.81. Figure 1 demonstrates two video frames classified by this model. These frames were not in the training set.



Figure 1. Video frames containing a combination of chimps, birds and humans. They have been classified by the model trained on YOLOv3.

These initial results are promising but the manual labelling of the training dataset is very time consuming. There are two solutions to this problem, the use of citizen science such as the Zooniverse project to gain access to a large base of human classifiers. Alternatively, source detection software can be used to label a large set of data which can then be refined by machine learning performance. For our future work, we are addressing both these options by creating a citizen science project to produce a large set of human classified training data as well as training models on labels produced by our source detection software.

References

- Burke, C., et al. 2018, International Journal of Remote Sensing (in press)
- Girshick, R. 2016, IEEE Transactions on Pattern Analysis and Machine Intelligence, 38
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems 25
- Longmore, S. N., et al. 2017, International Journal of Remote Sensing, 38, 2623. 1701.01611
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

MaxiMask: Identifying Contaminants in Astronomical Images using Convolutional Neural Networks

Maxime Paillassa,¹ Emmanuel Bertin,² and Hervé Bouy¹

¹*Laboratoire d'astrophysique de Bordeaux, Univ. Bordeaux, CNRS, B18N, allée Geoffroy Saint-Hilaire, 33615 Pessac, France,;*
maxime.paillassa@u-bordeaux.fr

²*Sorbonne Université, CNRS, UMR 7095, Institut d'Astrophysique de Paris, 98 bis bd Arago, 75014 Paris, France.*

Abstract. We present MaxiMask, a contaminant detector for ground-based astronomical images based on convolutional neural networks (CNNs). Once trained, MaxiMask is able to detect cosmic rays, hot pixels, bad pixels, saturated pixels, diffraction spikes, nebulous features, persistence effects, satellite trails and residual fringe patterns in ground based images, encompassing a broad range of ambient conditions, PSF sampling, detectors, optics and stellar density. Individual image pixels can be flagged through semantic segmentation, based on high-resolution probability maps generated by MaxiMask for each contaminant, except for the tracking error probability which is assigned by another dedicated CNN. Training and testing data have been gathered from a large dataset of simulated and real data originating from various modern CCD and near-IR cameras.

1. Introduction

Many astronomical studies rely on source catalogs extracted from digital images. However, defects, artifacts or contaminants present in the data can sometimes make these catalogs unreliable, especially when dealing with deep exposures. In this work we aim to mitigate this problem by identifying contaminated image pixels upstream so that these can be handled appropriately in subsequent analyses. Several attempts have been made to achieve this goal (e.g., Malapert & Magnard 2006; Desai et al. 2016). Existing tools generally require fine tuning and/or use algorithms dedicated to specific instruments/imaging surveys, which makes them unsuited to heterogeneous data gathered from public archives.

Convolutional neural networks (CNNs, LeCun et al. 1995) are supervised machine learning systems that can be trained to identify features in a wide variety of images, directly from the pixel data. Here, the system must be able to deal with various instruments, observation bands, stellar density regimes and pixel scales.

In the following, we present the data we have used for training; we then describe the CNN architecture and show an example of results obtained on test data.

2. The data

We choose to rely on real observations as much as possible. The training data comes from the COSMIC-DANCE survey (Bouy et al. 2013). It contains optical and near-infrared wide-field images from a wide variety of ground-based instruments. To build the training samples, our procedure is to add contaminants to “clean” images. These “clean” images are selected among CFHT-MegaCam, CTIO-Decam and HSC exposures for their superior cosmetics. Residual contaminants were eliminated by using each instrument pipeline.

Cosmic ray (CR) contaminants are extracted from COSMIC-DANCE dark images, fringe patterns (FR) from fringing maps, and nebulosities (NEB) from Herschel archives (Pilbratt et al. 2010). Hot columns/pixels (HC/HP) and bad columns/lines/pixels (BC/BL/BP) are simulated, as well as persistence effects (P), using a realistic model (Long et al. 2015), and satellite trails (STL), using SkyMaker (Bertin 2009). Other features are more tightly linked to the image content and must be directly identified within the “clean” exposures: saturated pixels (SAT), diffraction spikes (SP), bright background (BBG), background (BG). A training example is shown Fig. 1.

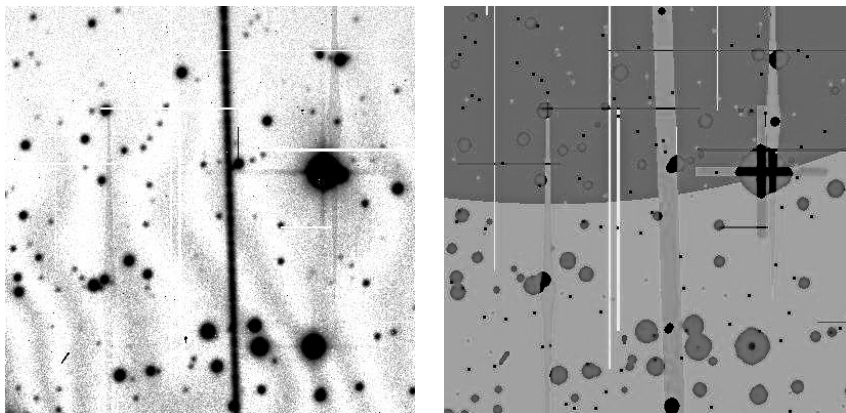


Figure 1. Example of an input image used for training (left), together with the contaminant ground truth (right, color-coded for illustration purposes). Red: CR, white: HC, yellow: BC, brown: BL, green: HP, blue: BP, turquoise: P, orange: STL, gray: FR, light gray: NEB, purple: SAT, light purple: SP, magenta: BBG, dark gray: BG. Black pixels are pixels affected by several classes.

3. The convolutional neural network

The CNN model used for semantic segmentation has a classical convolution-deconvolution architecture with VGG-like layers (Simonyan & Zisserman 2014). Upsampling takes advantage of the max-pooling indices of previous layers (Badrinarayanan et al. 2015) and includes two additional details (Yang et al. 2018) that allow the network to use the maximum of information at all resolution levels: (1) features computed in the downsampling convolution layers are added to the unpooled features from the upsam-

pling layers at identical resolution levels, and (2) pre-outputs from all resolution levels are concatenated to generate the final output (Fig. 2).

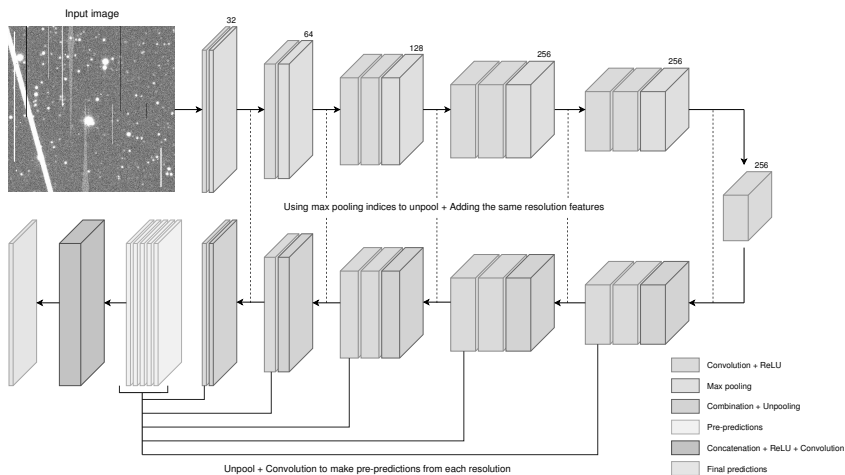


Figure 2. Architecture of the neural network

The model is trained to minimize the sigmoid cross-entropy (Rubinstein 1999) with ADAM optimization (Kingma & Ba 2014). The loss function incorporates a fraction of the pre-output cross-entropies computed at the three lowest resolution levels. To compensate for strong class imbalance, we apply a class-dependent weighting scheme to each pixel. Weight maps are smoothed with a 3×3 Gaussian kernel with $\sigma = 1$ to allow for a small margin around contaminants.

4. Results

Figure 3 shows qualitative results obtained from test data (not used for training) after training the model on a set of 50,000 400×400 sub-images.

5. Conclusion

Our CNN is able to reliably identify a large variety of contaminants over a wide range of astronomical exposures from ground-based imagers. MaxiMask is available on Github at <http://github.com/mpaillassa/MaxiMask>. Future improvements will likely include support for optical ghosts/reflections and saturation features in infrared instruments, as well as images coming from space-based instruments.

Acknowledgments. This research has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 682903, P.I. H. Bouy), and from the French State in the framework of the "Investments for the future" Program, IdEx Bordeaux, reference ANR-10-IDEX-03-02. This research has also received funding from the French

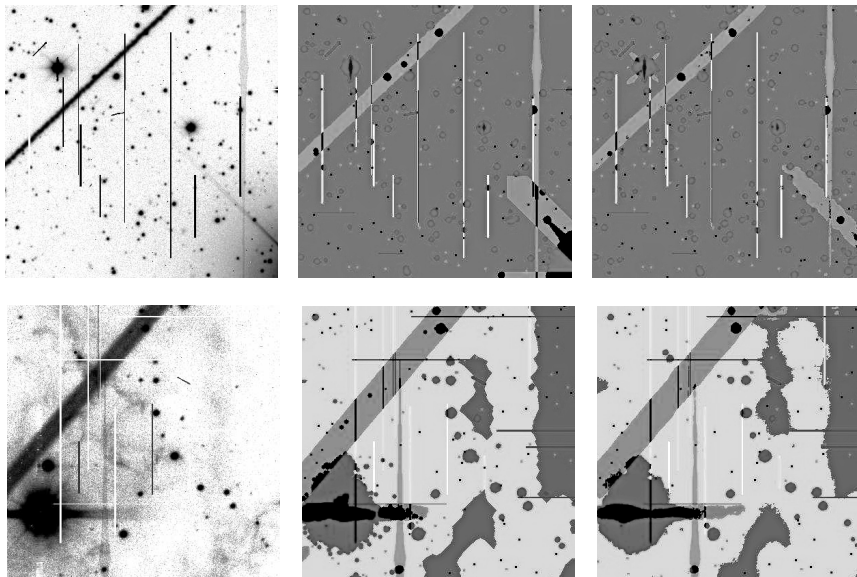


Figure 3. Left: original image; Center: contaminant map ground truth; Right: recovered contaminant map. A class is assigned to a pixel in the recovered map whenever the class probability exceeds the threshold that maximizes the Matthews correlation coefficient (Matthews 1975). Color coding is identical to that of Fig. 1

National Center for Space Studies (CNES). We gratefully acknowledge the support of NVIDIA Corporation with the donation of one of the Titan Xp GPUs used for this research.

References

- Badrinarayanan, V., Kendall, A., & Cipolla, R. 2015, arXiv preprint arXiv:1511.00561
- Bertin, E. 2009, *memsai*, 80, 422
- Bouy, H., Bertin, E., Moraux, E., Cuillandre, J.-C., Bouvier, J., Barrado, D., Solano, E., & Bayo, A. 2013, *A&A*, 554, A101. 1306.4446
- Desai, S., Mohr, J. J., Bertin, E., Kümmel, M., & Wetzstein, M. 2016, *Astronomy and Computing*, 16, 67. 1601.07182
- Kingma, D. P., & Ba, J. 2014, arXiv preprint arXiv:1412.6980
- LeCun, Y., Bengio, Y., et al. 1995, *The handbook of brain theory and neural networks*, 3361, 1995
- Long, K. S., Baggett, S. M., & MacKenty, J. W. 2015, *Persistence in the WFC3 IR Detector: an Improved Model Incorporating the Effects of Exposure Time*, Tech. rep.
- Malapert, J.-C., & Magnard, F. 2006, in *Astronomical Data Analysis Software and Systems XV*
- Matthews, B. W. 1975, *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405, 442
- Pilbratt, G. L., et al. 2010, *A&A*, 518, L1. 1005.5331
- Rubinstein, R. 1999, *Methodology and computing in applied probability*, 1, 127
- Simonyan, K., & Zisserman, A. 2014, arXiv preprint arXiv:1409.1556
- Yang, T., Wu, Y., Zhao, J., & Guan, L. 2018, *Cognitive Systems Research*

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

A Hybrid Neural Network Approach to Estimate Galaxy Redshifts from Multi-Band Photometric Surveys

Rafael Duarte Coelho dos Santos,¹ Felipe Carvalho de Souza,¹
Amita Muralikrishna,^{1,2} and Walter Augusto dos Santos Junior¹

¹*INPE - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brazil*

²*IFSP - Instituto Federal de São Paulo, Campus São José dos Campos, SP, Brazil*

Abstract. Machine learning methods have been used in cosmological studies to estimate variables that would be hard or costly to measure precisely, like, for example, estimating redshifts from photometric data. Previous work showed good results for estimating photometric redshifts using nonlinear regression based on an artificial neural network (MultiLayer Perceptron or MLP). In this work we explore a hybrid neural network approach that uses a Self-Organizing Map (SOM) to separate the original data into different groups, then applying the MLP to each neuron on the SOM to obtain different regression models for each group. Preliminary results indicate that in some cases better results can be achieved, although the computational cost may be increased.

1. Introduction

Formation of the Universe is studied through mapping its observed objects into different categories (e.g. by separating galaxies from stars) and by observing their spatial distributions, shapes and positions. For this redshifts can be used to estimate distances between galaxies and our own.

Photometry techniques have been used as an alternative estimate galaxies' redshifts. Information can be estimated about more objects, although with less accuracy. In this paper we explore a two-level neural network approach to estimate photometry redshifts, or photo-z.

2. Spectroscopic and Photometric Redshifts

Galaxy redshifts can be used as a distance measure and can be used to calculate the galaxy spatial distribution. The more accurate way to obtain the redshift is through the spectra of a galaxy via what is called spectroscopic redshift (Santos 2012). However spectroscopy has some limitations: one of them is that the observed objects should have enough brightness that they can be detected. This limits large surveys (LINEA 2017).

Photometry is a technique that was taken as an alternative to the spectroscopy in those cases when there is a need to collect data from many objects in a survey. It provides a quick approximation of the SED (Spectral Energy Distribution), with less

accuracy, and applicable to more objects. One way to do this is through regression using neural networks (Santos 2012) (Muralikrishna et al. 2019).

3. Data

For our tests, we used photometric information about 100,000 galaxies obtained from the SDSS survey (Abolfathi et al. 2017), which has the respective spectroscopy redshifts also recorded in a column so we can compare the results of our approach with the reference value.

The initial data was further processed: only records with guaranteed photometric quality were kept, and as in the previous work, records with redshift values lower than 0.01 and higher than 1 and with redshift errors values lower than 0 (zero) and larger than 0.0005 were removed. The pre-processed subset have around 50,000 records with six columns: one for each photometric band - u, g, r, i and z and the spectroscopic redshift value.

4. Neural Networks

In our previous work we used a MLP with a supervised training algorithm (Fausett 1994) to estimate the spectroscopic redshift from the photometric data through regression. Results were promising, but we consider that multiple MLPs, applied to different subsets of the data, may yield better results for the estimation of the regression coefficients.

In this paper we explore a two-level approach: a SOM (Kohonen 2011, Self-Organizing Map) which “splits” the data into different sets, and one MLP trained and applied to each set. Figures 1 and 2 illustrates our approach. Figure 1 shows the UGRIZ values for all the 50,000 records in a Parallel Coordinates plot (Inselberg 2009) – we can notice the variation between the values in the data set.

Figure 2 shows a set of Parallel Coordinates plots based on a Kohonen SOM, which each subplot corresponding to a SOM neuron that “captured” data points that were similar to data in the same neuron and somehow different from data in other neurons. All data from Figure 1 is distributed in the cells/neurons in Figure 2, but the cells’ values are less spread.

We then apply the same MLP neural network to each of the data’s subset defined by the SOM. We theorize that this second-level neural network will converge quicker and yield better results than our previous approach.

For each neuron on the SOM, a MLP was created and trained, with six different architectures (5, 10, 30, 50, 70 and 90 neurons on the hidden layer). The metric used during the training to measure the accuracy of the results was the Normalized Median Absolute Deviation (σ_{NMAD}) (Molino et al. 2017), calculated as shown in Equation 1):

$$\sigma_{NMAD} = 1.48 \times \text{median} \left(\frac{|\delta_z - \text{median}(\delta_z)|}{1 + z_s} \right) \quad (1)$$

where z_b is the photometric redshift, z_s is the spectroscopic redshift and $\delta_z = (z_b - z_s)$.

Each MLP was trained 10 times and the σ_{NMAD} stored for each run. The σ_{NMAD} are shown in box plots, in the same visual arrangement as the SOM neurons, in Figure 3.

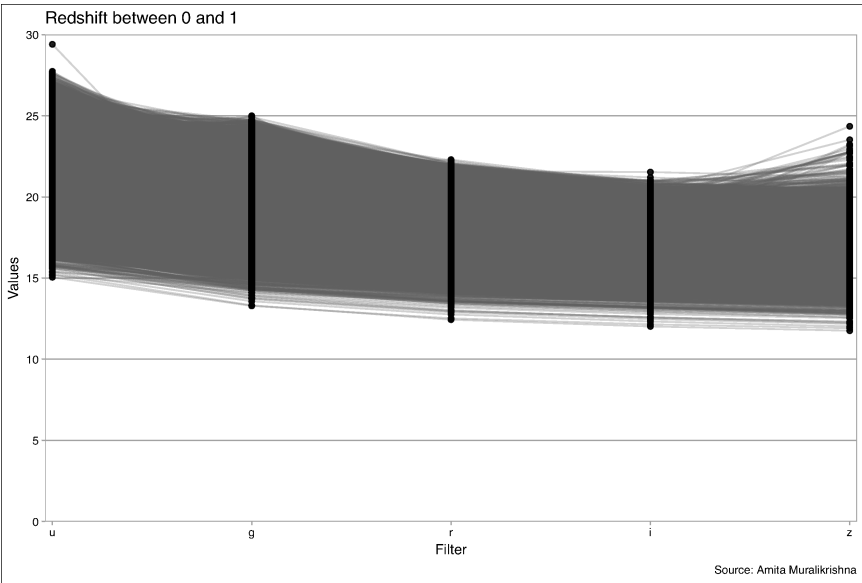


Figure 1. 50.000 UGRIZ data points.

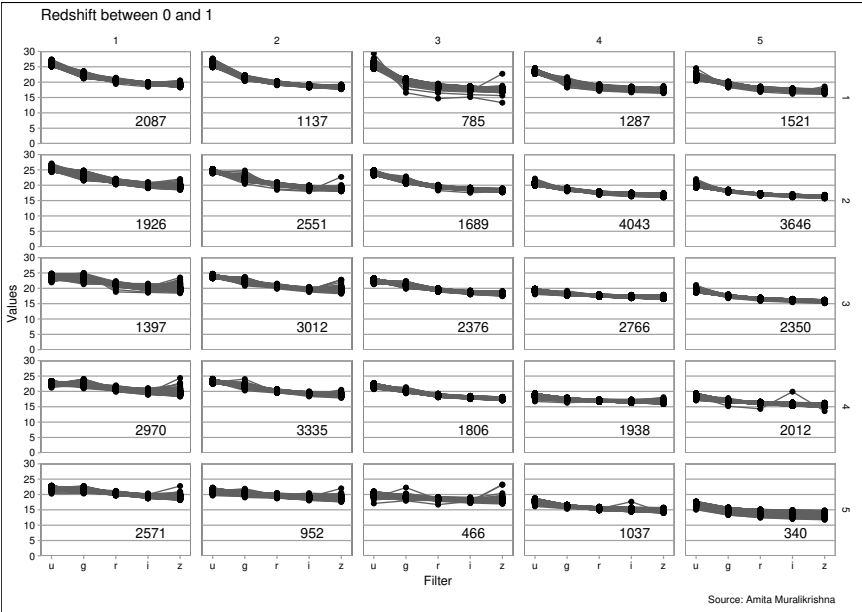


Figure 2. 50.000 UGRIZ data points, split in 5x5 subsets

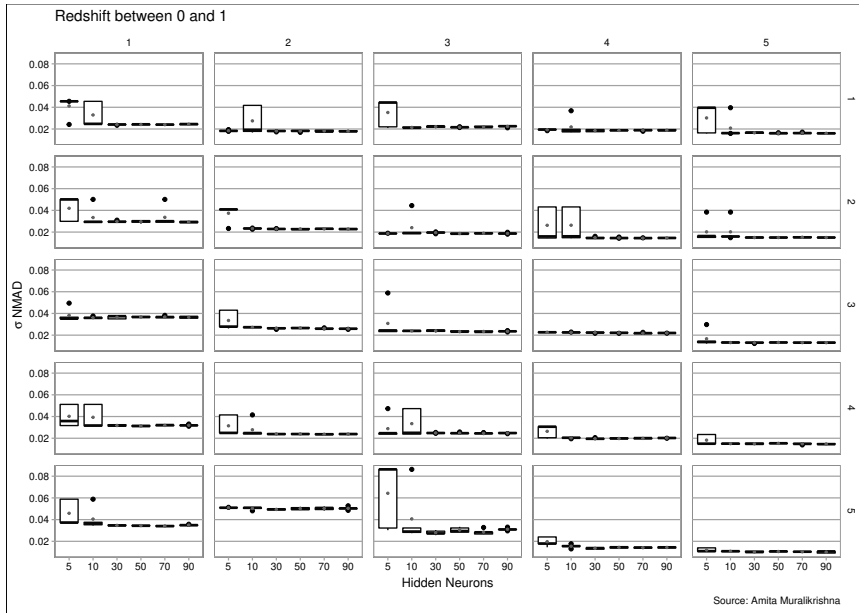


Figure 3. Box plots for σ_{NMAD} for each of the SOM neurons

5. Evaluation and Conclusion

From Figure 3 we can see that many of the data subsets had smaller σ_{NMAD} values than we obtained in our previous work (0.022 with 90 neurons in the hidden layer) – some cells (eg. C3R5, C5R5) presented good results ($\sigma_{NMAD} = 0.009$) even with a smaller number of neurons in the second level MLP. On the other hands, some of the cells (ex. C2R5) presented worse results than the obtained before.

We consider that our approach was able to split the data set into “easy” and “hard” subsets. “Easy” ones can be used for estimation of photo-z, and “hard” ones can be further explored with different machine learning algorithms.

References

- Abolfathi, B., et al. 2017, The Astrophysical Journal Supplement Series
- Fausett, V. L. 1994, Fundamentals of Neural Networks: Architectures, Algorithms and Applications (PrenticeHall)
- Inselberg, A. 2009, Parallel Coordinates – Visual Multidimensional Geometry and Its Applications (Springer)
- Kohonen, T. 2011, Self-Organizing Maps (Springer), 3rd ed.
- LINEA 2017, Laboratório Interinst. de e-Astronomia. URL <http://www.linea.gov.br>
- Molino, A., et al. 2017, Monthly Notices of the Royal Astronomical Society, 470, 95
- Muralikrishna, A., et al. 2019, in ADASS XXVII, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 405
- Santos, W. A. 2012, Ph.D. thesis, Instituto de Astronomia, Geofísica e Ciências Atmosféricas (IAG), Departamento de Astronomia, Universidade de São Paulo (USP)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Chatting with the Astronomical Data Services

André Schaff,¹ Alexis Guyot,² Thomas Boch,¹ and Sébastien Derriere¹

¹*Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg,
 UMR 7550, F-67000 Strasbourg, France; andre.schaff@astro.unistra.fr*

²*IUT, dijon, France*

Abstract. In our everyday life, we use increasingly the voice to interact with assistants for heterogeneous requests (weather, booking, shopping). We present our experiments to apply the Natural Language Processing (NLP) to the querying of astronomical data services. It is of course easy to prototype something, but is it realistic to propose it as a new way of interaction in a near future, as an alternative to the traditional forms exposing parameter fields, check boxes? To answer to this question, it is necessary to answer before to the most fundamental question: is it possible to satisfy professional astronomers needs through this way? We have not started from scratch as we have useful tools and resources (Sesame name resolver, authors in Simbad, missions and wavelengths in VizieR, etc.) and the Virtual Observatory (VO) (<http://www.ivoa.net/>) brings us standards (TAP, UCDs, etc) implemented in the CDS services. The interoperability, enabled by the VO, is a mandatory backbone. We explain how it helps us to query our services in natural language and how it will be possible in a further step to query the whole VO through this way. Our approach is pragmatic, based on a chatbot interface (involving Machine Learning) to reduce the gap between good and imprecisely/ambiguous queries.

1. Introduction

We did not start from scratch. We have useful tools and resources (Simbad name resolver, authors in Simbad, missions and wavelengths in VizieR, etc.) and the Virtual Observatory brings us standards (like TAP/ADQL, UCDs (Kou et al. 2005), etc) implemented in the CDS services. We have defined a first list of typical queries (Fig. 1). The translation from natural language to SQL has been explored at many times in the past (Giordani & Moschitti 2010; Zhong et al. 2017) and it provides good examples for its translation to ADQL.

What is the effective temperature of Sirius?

What are the galactic coordinates of Geminga?

Which galaxy interacts with NGC 4038?

Show me an image of the Pleiades in the K band

How many QSOs are there at redshift larger than 6? How many QSOs are there at $z > 6$?

What is the redshift of galaxies members of the Virgo cluster?

Figure 1. Example of typical queries in natural language

2. The user interface

The chatbot has a user interface (Fig. 2) written in Javascript which manages the design and the link with the CDS services. The natural language recognition is delegated to a customized Dialogflow agent. The user writes a request in natural language. The request sentence is then analyzed by the agent which returns its structure. We use then this structure to translate it in one or more queries (Fig. 3) to our services. In the last step, we dress the results.

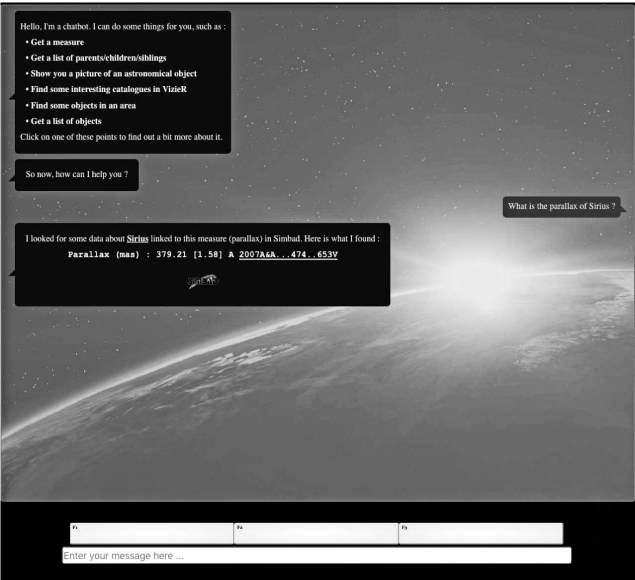


Figure 2. Chatbot user interface

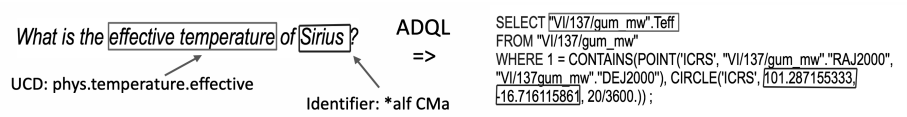


Figure 3. Structure of a question and its translation to ADQL

3. The natural language recognition

We made a first experiment with Stanford NLP libraries (<https://nlp.stanford.edu/software/>) but it required too much time to improve it. In a second step we chose Dialogflow (<https://dialogflow.com/>) to manage the recognition side. With Dialogflow we delegate a greater part of the tool than with the first approach and we can thus spend more time to better the quality of the user query translation. The link to the services and de facto to the data is the key part and our development is designed to be

able to switch to other NLP tools. The Dialogflow agent must be customized (intents, etc.; Fig. 4) and we evaluate the user requests (machine learning).

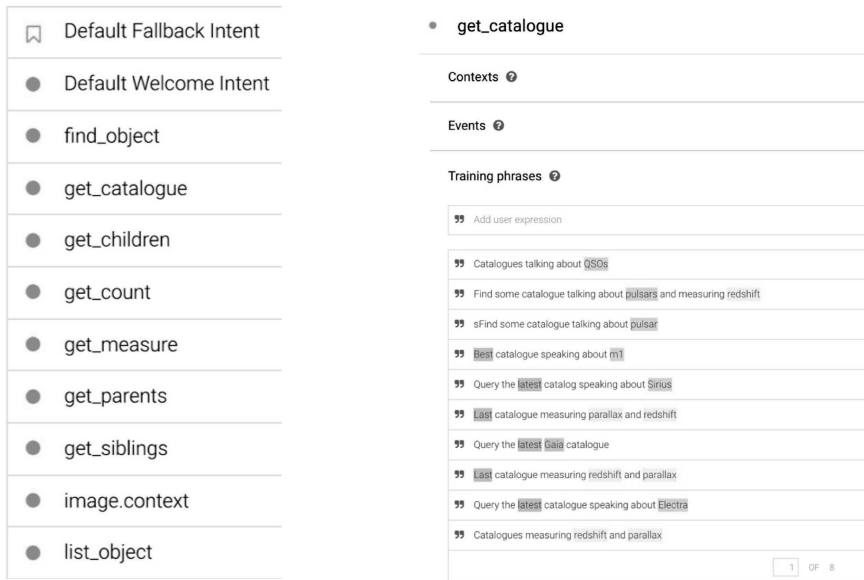


Figure 4. Intents and get_catalogue example

4. Status

The interoperability standards, enabled by the IVOA, are a mandatory backbone providing us a part of the mechanism useful to translate a natural language request to a query understandable by our services. The first presentations of the prototype (including widgets Fig. 5) encouraged us to continue and to improve the fitting of the results with the astronomers requests.

5. Next steps

- Open it to the community to improve it.
- More sophisticated chat with the user to refine a request.
- Evaluate the extension to other VO services, outside the CDS.
- Add voice recognition.

6. Perspectives

There are many potential applications.

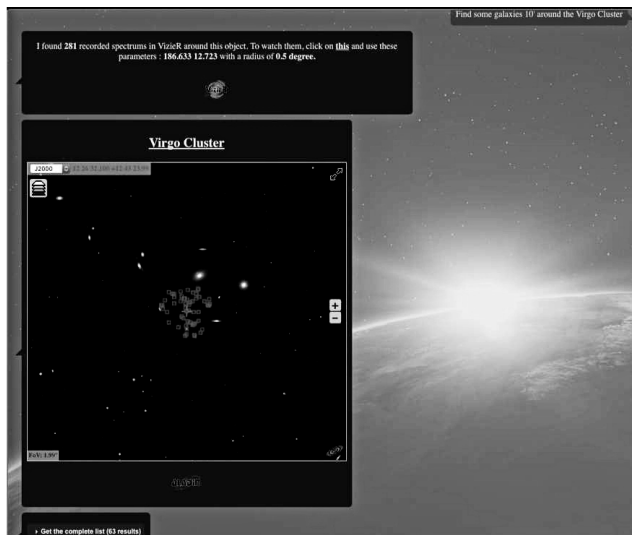


Figure 5. Aladin Lite as a widget

- The most important challenge is probably to integrate it into the CDS portal (<http://cdsportal.u-strasbg.fr/>), provided that sufficient accuracy and quality criteria are reached.
- In the field of education we plan to couple it to a virtual reality device which are now affordable (Schaaff & Polsterer 2017) (for example, to browse HiPS (Schaaff et al. 2015)) to offer direct voice control.

References

- Giordani, A., & Moschitti, A. 2010, in *Natural Language Processing and Information Systems*, edited by H. Horacek, E. Métais, R. Muñoz, & M. Wolska (Berlin, Heidelberg: Springer Berlin Heidelberg), 207
- Kou, H., Napoli, A., & Toussaint, Y. 2005, in *Natural Language Processing and Information Systems*, edited by A. Montoyo, R. Muñoz, & E. Métais (Berlin, Heidelberg: Springer Berlin Heidelberg), 32
- Schaaff, A., Berthier, J., Da Rocha, J., Deparis, N., Derriere, S., Gaultier, P., Houpin, R., Normand, J., & Ocvirk, P. 2015, in *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*, edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of *Astronomical Society of the Pacific Conference Series*, 125
- Schaaff, A., & Polsterer, K. L. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of *Astronomical Society of the Pacific Conference Series*, 663
- Zhong, V., Xiong, C., & Socher, R. 2017, *ArXiv e-prints*. 1709.00103

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Machine Learning from Cosmological Simulations to Identify Distant Galaxy Mergers

Gregory F. Snyder

Space Telescope Science Institute, Baltimore, MD, USA; gsnyder@stsci.edu

Abstract. I describe efforts to blend cosmological simulations with surveys of distant galaxies. In particular, I will discuss our work to create and interpret millions of synthetic images derived from the Illustris project, a recent large hydrodynamic simulation effort. Recently, we showed that because galaxies assembled so rapidly, distant mergers are more common than the simplest arguments imply. Further, we improved image-based merger diagnostics by training many-dimensional ensemble learning classifiers using the simulated images and known merger events. By applying these results to data from the CANDELS multi-cycle treasury program, we measured a high galaxy merger rate in the early universe in broad agreement with theory, an important test of our cosmological understanding.

1. Introduction

For almost a century, scientists have studied the composition, structure, and evolution of galaxies. Under the Λ CDM paradigm, galaxies are the visible end result of the gravitational collapse of dark matter over-densities, from fluctuations randomly seeded within the first instant. In and around galaxies, this assembly couples with magnetohydrodynamics and the physics of stars, black holes, and dust to complicate the picture of why galaxies appear the way they do. Exponential growth in computing power has given researchers the ability to test theories for galaxy evolution with ever-better numerical simulations and ultimately to clarify this picture. Achieving this goal requires carefully interpreting simulation results in the context of observations. Along the way, this interpretation should lead us to new insights about what our galaxy data says about their assembly over cosmic time.

An open question is what is the best methodology to achieve a useful blend of observations and simulations to understand galaxy formation? It is feasible to directly compare and contrast summary statistics (e.g., Bahé et al. 2016; Rodriguez-Gomez et al. 2018), primarily as a means to evaluate the goodness of match between theory and data and learn about our progress in understanding galaxy physics. Alternatively, one can measure the observability of galaxy features as a way to measure the intrinsic physical rate of certain events, such as galaxy mergers (e.g., Lotz et al. 2008; Snyder et al. 2017)

A possible approach to enhance our understanding of galaxy formation is to combine simulations and data with machine learning techniques. Historically, the procedures above fail to account for either the full detail of individual galaxy assembly histories or the full diversity of galaxy evolution pathways. In principle, large cosmological simulations (Schaye et al. 2014; Dubois et al. 2014; Pillepich et al. 2018) provide both, but the statistical methods we apply to them often gives us only a limited view of their

implications. By contrast, a machine learning approach can treat the physical evolution of individual galaxies as the inputs or “labels”, and it treats the observable features as the output or “data”. By training such a system to identify the inputs given the data, we can use galaxy surveys to measure directly the physical evolution of galaxies, in as full generality as the simulations allow. Applying this approach, Huertas-Company et al. (2018) determined how to identify a particularly interesting phase of distant galaxy evolution in HST data. In this contribution, we discuss a similar approach to the problem of identifying distant galaxy merger events (Figure 1; Snyder et al. 2018).

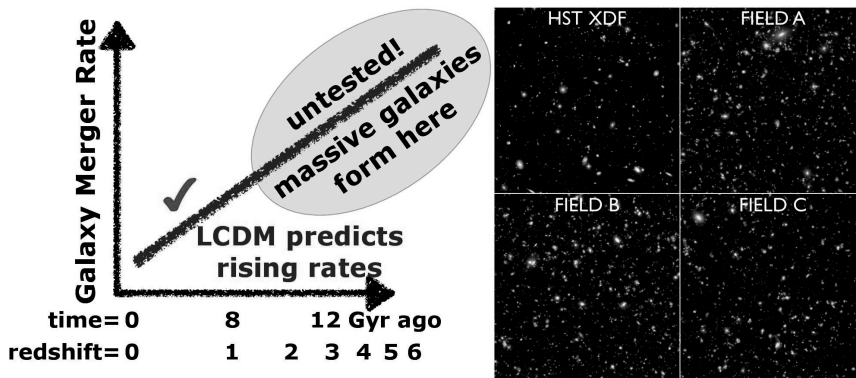


Figure 1. Although models predict a steeply rising merger rate, only recent data has reliably probed galaxy mergers in the first few billion years. Large cosmological simulations provide us the ability to mimic galaxy surveys (right, <https://doi.org/10.17909/T98385>), and given the known simulated history of galaxy mergers, to train new techniques for measuring galaxy mergers from real data.

2. Methods

We use two types of input data from the Illustris cosmological simulation: merger definitions and mock images. For this work, the merger definition is a 10:1 stellar mass ratio merger completing within ± 250 Myr (Rodríguez-Gomez et al. 2015) of the timestep of the mock image. This allows us to assign a single boolean value to each mock observation as the input merger label. Pristine synthetic images derive from an HST-funded archival project (HST-AR-13887) to produce and disseminate Illustris mock images to the community (Torrey et al. 2015). From these, we create $\approx 10^6$ realistic mock HST & JWST images with PSF and noise effects, mainly at $z > 1$, and measure common non-parametric morphology statistics (<http://www.illustris-project.org/data>). Figure 1 presents some related mock data products from <https://doi.org/10.17909/T98385>. See Snyder et al. (2018) for full details.

Using these labels and predictors, we trained 10-dimensional random forests (RFs) using five morphology statistics from each of I and H band mock images.

3. Results

We found that RFs produce superior merger classifications compared to individual morphology statistics, perhaps unsurprising given their utilization of more information. Importantly, all manual encoding methods we tried resulted in poor purity (30-50%) in Illustris. Thus, the simulation might be saying the morphology approach is harder than we thought, motivating future work with more comprehensive auto-encoding methods such as Deep Learning. In any case, it is important to keep this limitation in mind when interpreting distant galaxy morphology data. Even still, the RFs improve the statistics available for distant morphology-selected mergers. We apply the trained RFs to CANDELS HST catalogs, and find that RFs select bulges in post-mergers and asymmetries in pre-mergers. We find rising merger rate, similar to those inferred from pair statistics, but the RFs estimate 2X too many mergers (Figure 2). This may be caused by a mismatched morphology distributed in the simulation compared to reality, likely caused by imperfect galaxy astrophysics such as supernova feedback. Thus, techniques such as transfer learning might be necessary to improve upon these results.

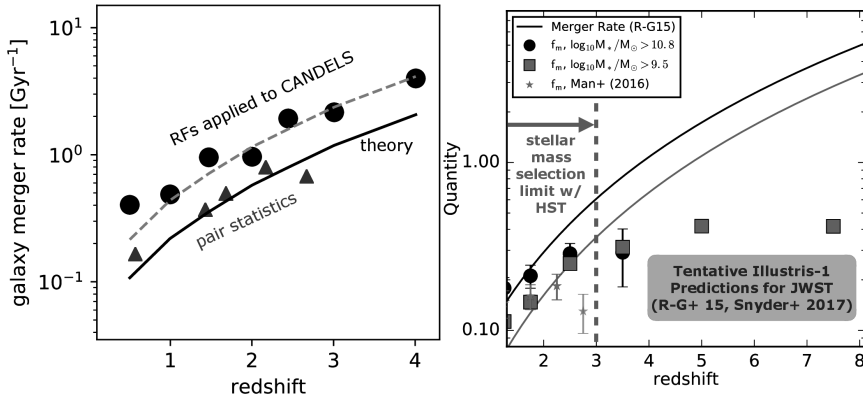


Figure 2. Results from Snyder et al. (2018, left) and predictions for JWST following Snyder et al. (2017, right). By blending cosmological simulations with future datasets, we hope to determine how galaxy building blocks assembled into the later populations of galaxies we observe.

4. Summary

1. We mock-observed Illustris in popular HST & JWST filters, and measured common non-parametric morphology statistics.
2. Using this manual encoding, we trained 10-d random forests on simulated mergers with a broad definition of 10:1 mass ratio within ± 250 Myr.
3. The RFs achieve improved completeness by leveraging different features in pre-mergers versus post-mergers.
4. Applying the RFs to data, we recover the expected rise in merger rates versus redshift, matching data from earlier merger stages (pairs).

References

- Bahé, Y. M., Crain, R. A., Kauffmann, G., Bower, R. G., Schaye, J., Furlong, M., Lagos, C., Schaller, M., Trayford, J. W., Vecchia, C. D., & Theuns, T. 2016, *MNRAS*, 456, 1115. URL <https://academic.oup.com/mnras/article-lookup/doi/10.1093/mnras/stv2674>
- Dubois, Y., Pichon, C., Welker, C., Le Borgne, D., Devriendt, J., Laigle, C., Codis, S., Pogosyan, D., Arnouts, S., Benabed, K., Bertin, E., Blaizot, J., Bouchet, F., Cardoso, J.-F., Colombi, S., de Lapparent, V., Desjacques, V., Gavazzi, R., Kassir, S., Kimm, T., McCracken, H., Milliard, B., Peirani, S., Prunet, S., Rouberol, S., Silk, J., Slyz, A., Sousbie, T., Teyssier, R., Tresse, L., Treyer, M., Vibert, D., & Volonteri, M. 2014, *MNRAS*, 444, 1453. URL <http://adsabs.harvard.edu/abs/2014MNRAS.444.1453D>
- Huertas-Company, M., Primack, J. R., Dekel, A., Koo, D. C., Lapiner, S., Ceverino, D., Simons, R. C., Snyder, G. F., Bernardi, M., Chen, Z., Domínguez-Sánchez, H., Chen, Z., Lee, C. T., Margalef-Bentabol, B., & Tuccillo, D. 2018, eprint arXiv:1804.07307. URL <http://arxiv.org/abs/1804.07307>
- Lotz, J., Jonsson, P., Cox, T., & Primack, J. 2008, *MNRAS*, 391, 1137
- Pillepich, A., Springel, V., Nelson, D., Genel, S., Naiman, J., Pakmor, R., Hernquist, L., Torrey, P., Vogelsberger, M., Weinberger, R., & Marinacci, F. 2018, *MNRAS*, 473, 4077. URL <http://academic.oup.com/mnras/article/473/3/4077/4494369>
- Rodriguez-Gomez, V., Genel, S., Vogelsberger, M., Sijacki, D., Pillepich, A., Sales, L. V., Torrey, P., Snyder, G., Nelson, D., Springel, V., Ma, C.-P., & Hernquist, L. 2015, *MNRAS*, 449, 49. URL <http://adsabs.harvard.edu/abs/2015MNRAS.449...49R>
- Rodriguez-Gomez, V., Snyder, G. F., Lotz, J. M., Nelson, D., Pillepich, A., Springel, V., Genel, S., Weinberger, R., Tacchella, S., Pakmor, R., Torrey, P., Marinacci, F., Vogelsberger, M., Hernquist, L., & Thilker, D. A. 2018, *MNRAS*. URL <https://academic.oup.com/mnras/advance-article/doi/10.1093/mnras/sty3345/5237717>
- Schaye, J., Crain, R. A., Bower, R. G., Furlong, M., Schaller, M., Theuns, T., Dalla Vecchia, C., Frenk, C. S., McCarthy, I. G., Helly, J. C., Jenkins, A., Rosas-Guevara, Y. M., White, S. D. M., Baes, M., Booth, C. M., Camps, P., Navarro, J. F., Qu, Y., Rahmati, A., Sawala, T., Thomas, P. A., & Trayford, J. 2014, *MNRAS*, 446, 521. URL <http://adsabs.harvard.edu/abs/2015MNRAS.446...521S>
- Snyder, G. F., Lotz, J. M., Rodriguez-Gomez, V., da Silva Guimarães, R., Torrey, P., & Hernquist, L. 2017, *MNRAS*, 468, 207. URL <http://dx.doi.org/10.1093/mnras/stx487>
- Snyder, G. F., Rodriguez-Gomez, V., Lotz, J. M., Torrey, P., Quirk, A. C. N., Hernquist, L., Vogelsberger, M., & Freeman, P. E. 2018. URL <http://arxiv.org/abs/1809.02136>
- Torrey, P., Snyder, G. F., Vogelsberger, M., Hayward, C. C., Genel, S., Sijacki, D., Springel, V., Hernquist, L., Nelson, D., Kriek, M., Pillepich, A., Sales, L. V., & McBride, C. K. 2015, *MNRAS*, 447, 2753. URL <http://adsabs.harvard.edu/abs/2015MNRAS.447.2753T>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

A Machine Learning Approach for Dark-Matter Particle Identification Under Extreme Class Imbalance

Raymond Sutrisno, Ricardo Vilalta, and Andrew Renshaw

University of Houston, 4800 Calhoun Rd., Houston TX, USA;
rasutrisno@uh.edu

Abstract. The Darkside-50 collaboration is an international experiment conducted at the Laboratori Nazionali del Gran Sasso in Italy, where low-radioactivity liquid argon is used within a dual-phase time projection chamber to detect weakly interacting massive particles (WIMPs), one of the leading candidates for dark matter. The Darkside-50 experiment faces two main data-analysis challenges: extreme class imbalance and large datasets. In this paper we show how machine learning techniques can be employed, even under the presence of samples exhibiting extreme class-imbalance (i.e., extreme signal-to-noise ratio). In our data-analysis study, the ratio of negative or background events to positive or signal events is highly imbalanced by a factor of 10^7 . This poses a serious challenge when the objective is to identify a signal that can be easily misclassified as background. We compare several techniques in machine learning that deal with the class imbalance problem: ROUS, SMOTE, and MSMOTE. Experimental results on real data obtained from the Darkside-50 experiment show very high recall values (~ 0.985), with reasonable performance in terms of precision (~ 0.80) and F1-score (~ 0.875).

1. Introduction

Many candidates have been hypothesized to describe the apparent missing matter in the Universe, all of which would fall under the category of dark matter. It is referred as dark matter because it is non-luminous and direct observation using traditional astronomical techniques is not possible. Instead, its presence has been inferred by its gravitational effect on surrounding luminous matter, as well as the footprints it has left within the cosmic microwave background throughout the history of the Universe. Among the leading hypothesized candidates, weakly interacting massive particles (WIMPs) have become a favorite for experimentalists, since their interaction with normal matter can be predicted and searched in ultra-sensitive detectors. Interacting via only the weak force, a WIMP particle would have the potential of elastically scattering off the nucleus of an atom that is contained inside a detector here on Earth, producing what is called a nuclear recoil, and giving an avenue for the direct detection of a new particle that could explain the dark matter puzzle. These nuclear recoils would be detectable inside detectors such as the DarkSide-50 detector (Agnes et al. 2015), currently operating at the Laboratory Nazionali Gran Sasso in Italy. DarkSide-50 is an ultra-low background liquid argon time projection chamber built specially for the detection of a WIMP recoiling off the nucleus of an argon atom in the detector, and has been instrumental in the search for high- and low-mass WIMPs (Agnes et al. 2018).

The generated light signals coming from the nuclear recoil of a WIMP with the liquid argon inside the DarkSide-50 detector are captured by photomultiplier tubes set

in two arrays, at the top and bottom of the detector. Signals recorded by the photomultiplier tubes are used to reconstruct interactions. However, even with the ultra-low background levels in the DarkSide-50 detector, the expected rate of electromagnetic interactions inside the detector are quite large compared to the expected rate coming from WIMP nuclear recoils. These electromagnetic interactions result from beta-particles and gamma-rays interacting with the orbital electrons of the argon atoms in the detector, which give off a slightly different response relative to the nuclear recoil from a WIMP, allowing for the possibility to distinguish a WIMP signal from background events. The very low expected rate of WIMP interactions in DarkSide-50 (less than 10^{-2} per year), coupled with the background rate (order of 10^7 per year), creates an extreme class imbalance between the potential WIMP signal and the background within a data set that is quite large. With this in mind, an approach to classifying the data using machine learning has been explored and described next.

2. Machine Learning for Dark-Matter Particle Identification

In supervised learning or classification, we assume the existence of a training set of examples, $T = \{(\mathbf{x}_i, y_i)\}$, where vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is an instance of the input space \mathcal{X} , and y is an instance of the output space \mathcal{Y} . The output of the learning algorithm is a hypothesis (or function) $f(\mathbf{x})$ mapping the input space to the output space, $f: \mathcal{X} \rightarrow \mathcal{Y}$. In our case, vector \mathbf{x} corresponds to features characterizing events, including the following: the number of pulses detected, the integral of the first pulse (S_1), the integral of the second pulse (S_2), the position where the event was located within the detector (given by $\langle x_{pos}, y_{pos}, l_{drift} \rangle$), and the ratio of the integral of the first 90 nanoseconds of the first pulse relative to S_1 . The class y can be a Nuclear-Recoil (NR) event (positive example), or an Electron-Recoil (ER) events (negative example).

The Class Imbalance Problem. A common difficulty when applying supervised learning is the presence of tasks with highly imbalanced class priors (i.e., extreme signal-to-noise ratio). This is clearly the case when searching for dark matter particle interactions, where the ratio of background to signal events is highly imbalanced by a factor of 10^7 . The difficulty comes in identifying a signal that can be easily misclassified as background. Several techniques have been proposed to deal with the class imbalance problem (Japkowicz 2000). In general, many solutions rely on basic operations: undersampling the majority class, or oversampling the minority class (sampling is usually done under a uniform random distribution). We describe the techniques used in our experiments based on these basic operations.

Random Oversampling and UnderSampling Technique (ROUS). The first technique simply oversamples the minority class and undersamples the majority class. Sampling is done under a uniform distribution. The procedure continues until we reach a perfectly balanced class distribution. Oversampling the minority class is known to lead to overfitting, but such adverse scenarios can be reversed when the majority class is simultaneously undersampled.

Synthetic Minority Over-Sampling Technique (SMOTE). Rather than directly oversampling the minority class by creating copies of existing minority-class examples, SMOTE generates synthetic examples by creating new instances along the vectors connecting a minority-class example with the k -nearest-neighbors of the same class (Chawla et al. 2002). The SMOTE algorithm is parameterized by the number of nearest

neighbors K , and an integer N representing the percentage of newly generated synthetic examples. The rationale is to oversample the minority class by *spreading* the location of new examples on the input space; this helps to reduce model complexity (i.e., to avoid overfitting).

Modified SMOTE (MSMOTE). MSMOTE is a modified version of SMOTE that attempts to be more selective when oversampling. This is done by classifying neighbor examples into three categories: security, border, and noise. Whereas SMOTE would consider all three types, MSMOTE focuses on security examples only. The designation is determined by examining the classes of the k -nearest neighbors. If all k -nearest neighbors share the same class as the minority-class example under analysis, the example is considered of type security. A mixture of classes in the neighborhood of a minority-class example suggests the presence of noise or of examples at the border of minority and majority class regions; discarding these examples is hypothesized to improve performance.

3. Experimental Setting

We report on a set of experiments using real data from the DarkSide-50 detector. We tackle the class-imbalance problem using the techniques described above (ROUS, SMOTE, and MSMOTE). We consider Nuclear-Recoil (NR) events as positive examples and Electron-Recoil (ER) events as negative examples. We use Random Forests (Ho 1995) as the core learning algorithm. Random Forests and Decision Tree learners come from the Scikit-Learn Python Machine Learning library. We invoke the Classification and Regression Tree algorithm (CART; Breiman et al. 1984), with Gini as the splitting criterion.

Every result is the output of 30 training and testing runs. Both training and testing samples are obtained using stratified random sampling with ten thousand examples for training and fifteen thousand for testing. The huge amount of initial data ($\sim 2.4 \times 10^6$ ER events and 2.6×10^4 NR events) leads to data processing with high computational cost. Our experiments make use of a computer cluster with 5,704 CPU cores in 169 compute and 12 GPU nodes; cpu type is Intel Xeon E5-2680v4, with approx. 40TB of disk space and hundreds of GB in memory.

4. Results and Discussion

We use three performance metrics to assess model quality in the detection of Nuclear-Recoil events: precision, recall, and F1 score. The metrics are defined as a function of the number of true positive (tp), true negative (tn), false positive (fp), and false negative (fn) predictions. They are commonly used in classification tasks with skewed class distributions. The definitions are as follows:

$$\text{Recall} = \frac{tp}{tp+fn} \quad \text{Precision} = \frac{tp}{tp+fp} \quad \text{F1 score} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

Figure 1 contains plots of mean recall, precision, and F1 score when using the three techniques designed to handle skewed distributions. The x-axis corresponds to the amount of oversampling as a percentage of the number of minority-class examples (e.g., $N = 600$ means six additional examples are created for every original example

in the minority class). Results show high values of recall (~ 0.985) that stay relatively constant as oversampling grows; this is indicative of models bearing high sensitivity, where very few signal events are classified as background. Precision, on the other hand, shows lower performance (~ 0.80); it indicates many background events end up classified as signal events. This is expected considering the overwhelmingly small signal to noise ratio. But performance shows improvement as oversampling grows. F1 score (~ 0.875) is simply a harmonic mean of recall and precision and also shows improvement with increased oversampling. In terms of differences across techniques to handle skewed distributions, all of them perform similarly considering the amount of deviation around the mean (shadow regions in Figure 1). We conclude that a simple over- and under-sample technique suffices to handle the class imbalance problem in this particular domain, and that additional work is needed to avoid incorrectly classifying background events as signals.

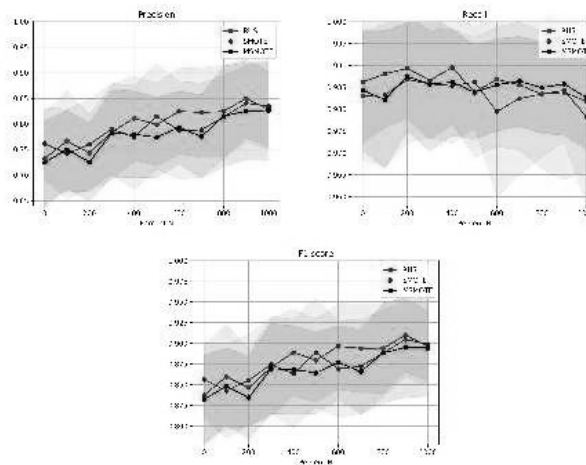


Figure 1. Recall, Precision and F1 score for ROUS, SMOTE, and MSMOTE. Background shadowed regions correspond to \pm one standard deviation estimated from thirty runs.

Acknowledgments. This work was partly supported by the Center for Advanced Computing and Data Systems (CACDS) at the University of Houston.

References

- Agnes, P., et al. (DarkSide) 2015, Phys. Lett., B743, 456. 1410.0653
- 2018, Phys. Rev. Lett., 121, 081307. 1802.06994
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. 1984, Wadsworth International Group
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002, Journal of Artificial Intelligence Research, 16, 321
- Ho, T. K. 1995, in Document analysis and recognition, 1995., proceedings of the third international conference on (IEEE), vol. 1, 278
- Japkowicz, N. 2000, in Proceedings of the International Conference on Artificial Intelligence ICAI00

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Analysis of Stellar Spectra from LAMOST DR5 with Generative Spectrum Networks

Rui Wang^{1,2,3} and A-Li Luo^{1,3}

¹*National Astronomical Observatories, Chinese Academy of Sciences, Beijing, China; wangrui@nao.cas.cn, lal@bao.ac.cn*

²*University of Chinese Academy of Sciences, Beijing 100049, China*

³*Key Laboratory of Optical Astronomy, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China*

Abstract. We derive the fundamental stellar atmospheric parameters (T_{eff} , $\log g$, $[\text{Fe}/\text{H}]$ and $[\alpha/\text{Fe}]$) of low-resolution spectroscopy from LAMOST DR5 with Generative Spectrum Networks (GSN), which follows the same scheme as a normal ANN with stellar parameters as the inputs and spectrum as outputs. After training on PHOENIX theoretical spectra, the GSN model performed effectively on producing synthetic spectra. Combining with a Bayes framework, application in analysis of LAMOST observed spectra becomes efficient on the Spark platform. Also, we examined and validated the results by comparing with reference parameters of high-resolution surveys and astero-seismic results. Our method is credible with a precision of 130K for T_{eff} , 0.15 dex for $\log g$, 0.13 dex for $[\text{Fe}/\text{H}]$ and 0.10 dex for $[\alpha/\text{Fe}]$.

1. Introduction

Most sky surveys result in extensive databases of stellar spectra for dissecting and understanding the Milky Way. The fundamental information derived from such spectra includes the effective temperature (T_{eff}), logarithm of surface gravity ($\log g$), abundance of metal elements with respect to hydrogen ($[\text{Fe}/\text{H}]$), and the abundance of alpha elements with respect to iron ($[\alpha/\text{Fe}]$), are valuable for Galactic archaeology and stellar evolution history. Many projects have been carried out to detect specific objects at high/low resolution covering a range of wavelengths.

In this report, we designed a new structure of artificial neural networks, Generative Spectrum Networks (GSN), a similar neural network proposed by Dafonte et al. (2016), which follows the same scheme as a typical ANN, except that the inputs and outputs are inverted. This approach was proposed and applied to simulations of prospective Gaia RVS (Cropper et al. 2018) spectra based on the Kurucz model (Kurucz 1993). However, real observed spectra were not tested. It should be noted that there is a sign discrepancy between the synthetic and observed spectra for errors from extinction, reddening, seeing, contamination of stray light, instruments and post data processing. We improve the generative artificial neural network training on Phoenix (Husser et al. 2013) spectra for estimation of the parameters of LAMOST (Luo et al. 2015) observations. In combination with a Bayesian framework and Monte Carlo (MC) method, the networks can be

used to derive not only stellar atmospheric parameters, but also their posterior distribution. The computing cost is always an insurmountable obstacle during the application of the MC method for a large number of data-sets. However, the distributed computing platform SPARK improves the viability of employing MC sampling methods based on Bayes theory. To the best of our knowledge, we are the first group to utilize SPARK estimating stellar parameters in this way. Moreover, our method adds an abundance of alpha elements ($[\alpha/\text{Fe}]$) with respect to the existing catalog provided by LASP (Luo et al. 2015; Wu et al. 2011).

2. Data

The spectra employed in this report consist of two parts: synthetic spectra calculated from PHOENIX mode (Husser et al. 2013) and LAMOST spectra from the internal fifth data release (LAMOST DR5; Luo et al. in preparation). The synthetic spectra with reference parameters (T_{eff} , $\log g$, $[\text{Fe}/\text{H}]$ and $[\alpha/\text{Fe}]$) are used for training and testing the GSN model. Then, the stellar parameters of the LAMOST spectra were estimated using the achieved GSN model.

3. Methods

In this report, we designed a new structure of artificial neural networks which consists of a fully-connected network with an input layer, three hidden layers, and an output layer, to generate spectra by training PHOENIX model spectra. Generative Spectrum Networks can produce model spectra when a team of parameters is given. Using chi-square distance as a proxy to match the spectrum to be measured with model grids is common and most methods use this approach. However, the uncertainty estimation would be difficult for template matching. Combined with Bayes rule, Monte Carlo sampling is an effective way to obtain the posterior distribution over the parameters given the observed spectrum.

4. Results

To ensure the reliability and accuracy of the stellar parameters obtained with GSN, we employed the parameter catalogs of some sub-sample catalogs of LAMOST DR5 common stars, with external precise stellar parameters derived from high-resolution observations, or by other methods used for comparing and validation. To obtain reliable results, we only selected spectra with $\text{SNR}_g > 30$ for comparison purposes.

The comparison of LAMOST DR5 with APOGEE DR14 (Holtzman et al. 2018) and PASTEL catalogue (Soubiran et al. 2016) are provided in the Fig 1 and Fig 2. As shown in the figures, our results of stellar parameters show great agreement with the results derived from the high-resolution APOGEE spectra and the results from other literatures.

References

- Cropper, M., Katz, D., Sartoretti, P., et al. 2018, A&A, 616, A5
 Dafonte, C., Fustes, D., Manteiga, M., et al. 2016, A&A, 594, A68. 1607.05954

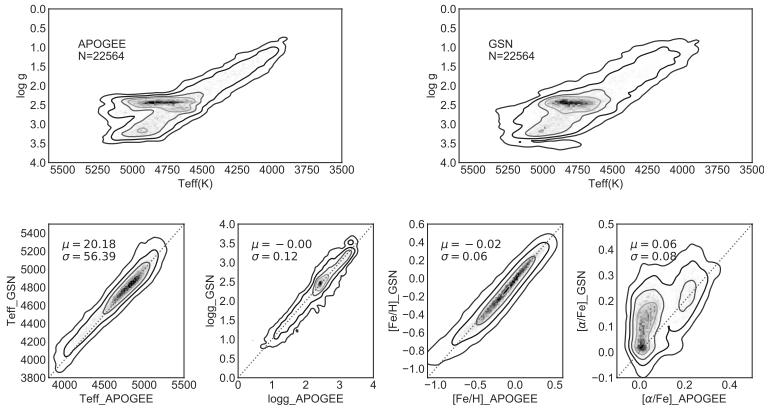


Figure 1. The density distribution of APOGEE parameters (top left panel) and GSN results (top right panel) for LAMOST 23,315 stars with SNR_g greater than 30. Also, the comparison between GSN stellar parameters and the APOGEE parameters are shown in the four bottom panels. The red dashed lines in the bottom panels are one-to-one lines.

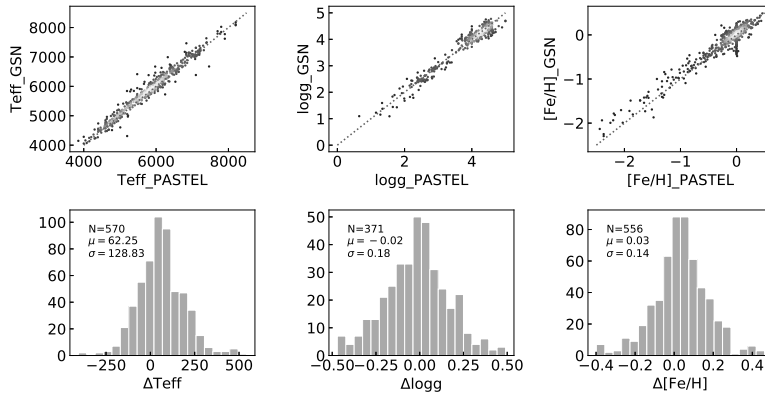


Figure 2. The color-coded scatter of GSN stellar parameters compared with the PASTEL catalogue for LAMOST DR5 spectra with $\text{SNR}_g \geq 30$ are shown in the three top panels. The red dash lines in the top panels are one-to-one lines. The distributions of the discrepancies for T_{eff} , $\log g$ and $[\text{Fe}/\text{H}]$ are shown in the three bottom panels.

Holtzman, J. A., Hasselquist, S., Shetrone, M., et al. 2018, AJ, 156, 125. 1807.09773
Husser, T.-O., Wende-von Berg, S., Dreizler, S., et al. 2013, A&A, 553, A6. 1303.5632
Kurucz, R. L. 1993, SYNTHE spectrum synthesis programs and line data
Luo, A.-L., Zhao, Y.-H., Zhao, G., et al. 2015, Research in Astronomy and Astrophysics, 15, 1095
Soubiran, C., Le Campion, J.-F., Brouillet, N., & others. 2016, A&A, 591, A118. 1605.07384
Wu, Y., Luo, A.-L., & Li, e., H.-N. 2011, Research in Astronomy and Astrophysics, 11, 924. 1105.2681



Mini Conference Photo (Photo: Peter Teuben)



Roberto Pizzo getting worried and Peter Teuben worries almost over (Photo: Keith Shortridge)

U-NetIM: An Improved U-Net for Automatic Recognition of RFIs

Min Long,¹ Zhicheng Yang,² Jian Xiao,² Ce Yu,² and Bo Zhang³

¹*Boise State University, Boise, United States of America*

²*Tianjin University, Tianjin, China; xiaojian@tju.edu.cn*

³*Chinese Academy of Sciences, Beijing, China*

Abstract. Radio frequency interference (RFI) mitigation is a key phase in data processing pipeline of radio telescopes. Classical RFI mitigation methods depending on the RFI physical characteristics can often fail to recognize some complicated patterns or result in misrecognition. We developed a novel approach of RFI recognition and automatic flagging using an improved convolution neural network. The improved U-Net model (U-NetIM) is constructed based on U-net with much deeper network structure for more complicated patterns and more components to reduce recognition error caused by over-fitting. The experiments show that the U-NetIM has better performance on both precision and recall rate than SumThreshold the most widely used classical method, U-Net, a traditional deep learning model and KNN, a typical machine learning model.

1. Introduction

Recognizing and marking the radio frequency interference (RFI) is a key step in the data processing of radio telescope observations. However, in the traditional recognizing process, manual intervention is often required, which greatly affects the efficiency of data processing. This paper aims to explore the application of deep learning technology in the automatic RFI recognition and provide a technical reference and application for the optimization of data processing systems adopted by large radio telescopes such as Five-hundred-meter Aperture Spherical radio Telescope (FAST).

RFI refers to any unwanted signal entering the radio telescope receiving system Mosiane et al. (2017) and has variety of manifestations. Some RFI are very scattered in spectrum, affecting a wide range of channels (e.g., wideband); some appear to be concentrated, affecting only certain channels (e.g., narrowband). Meanwhile, the RFI can be instantaneous, burst-like pulse (high intensity, short time), or it can occur continuously over a period of time, like standing waves (intensity changes periodically with time). Most RFIs are much stronger than common astronomical signals. If the received signals consist of vertical or horizontal envelopes of a wide or narrow band, (i.e, discrete or high-intensity occurrence), it is very likely to be contaminate by RFIs and causes misrecognition.

Current methods of RFI recognition can be mainly divided into the categories of linear detection, threshold-based algorithms, machine learning and deep learning methods. For the RFI with repetitive features in the time-frequency domain, like standing waves, the linear detection methods can achieve very good results. However, it fails in recognizing more complex signals, such as irregular signals generated by satellite

operation. The threshold-based algorithm has a good performance on discrete RFIs when the observation background is relatively stable. The SumThreshold Offringa et al. (2010) is the most widely used threshold-based algorithm in the existing radio data processing. However, when a very large number of RFIs are present, affecting most channels, the threshold-based algorithm will be less effective. The artificial intelligence methods represented by machine learning and deep learning have been widely used in image recognition, nature language processing, etc., and may can provide improvements for the automation and accuracy of RFI recognition.

2. Deep Learning Modeling

The recognition of RFI means to search for regions with significantly increased intensity or certain special features on the time-frequency plane that are similar to the image. The network model used to process image is mainly Convolutional Neural Network (CNN). The CNN is named after the convolutional operation that multiplies all the values in the region covered by the convolutional kernel and then adds them together. The shallow layers extract the texture information of the image. And then the deep layers integrate the features from the shallow layers for the semantic information. The Pooling layers are used to filter the information. Finally, through multiple convolutions and pooling layers, specific information about certain areas of the image is obtained and then marked.

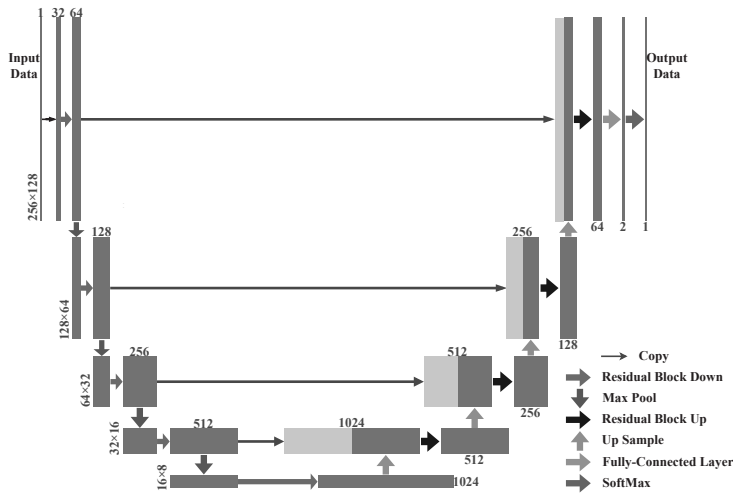


Figure 1. Network architecture of the proposed U-NetIM model. Blue boxes represents multi-channel feature map with the number of channels denoted on the top and the x-y-size to the left. The arrows denote the flow of operations.

This paper proposed an improved U-NetRonneberger et al. (2015) model (U-NetIM) of RFI recognition considering the features of the RFI discussed above. More layer are added to the network to obtain more information form the data for complex RFI.

For the narrow band RFIs, they challenge more precise searching and thus small convolution kernels of size 3 are used. As for wide band RFI, multi-layer convolution is used to expand the convolution field of view. Since RFI appears intensely, while the background is relatively stable, it is very suitable for Max-pooling to retain more changed information. Moreover, the random RFI may make the data distribution vary greatly. Therefore Batch-Normalization is used to scale the data to make the distribution more stable for better recognition performance.

The proposed U-NetIM model is shown in Fig. 1. The left side of the model is the down-sampling path, the data enters from the upper left corner, and the result of each operation (colored arrow) is subjected to the Max-pooling operation. Meanwhile the results are sent to the corresponding layer in the up-sampling path as copies. After four operations, the data comes to the bottom of the model, which is followed by the up-sampling path on the right. The up-sampling path is combined with the information extracted from the down-sampling path to gradually mark the RFIs on the original data. After the up-sampling path, the data will reach the upper right corner with the same size as the input data. After that, the channels of data are reduced to 2 via a fully connected layer and Softmax operation, which represents the the RFI category corresponding to each pixel. Finally, the category is taken as the output result.

3. Experiments and Results

This paper tested the U-NetIM model and compared the results with traditional methods, such as U-Net, KNN, and SumThreshold, in aspect of the accuracy and efficiency of RFI recognition. The experiment used simulation data generated by HIDE Akeret et al. (2017). The data set is divided into astronomical data as input data and Ground_Truth as standard recognition results. The Loss is calculated from the output data produced from neural network and the Ground_Truth. The data set consists of 2 parts, 2900 training data and 76 test data.

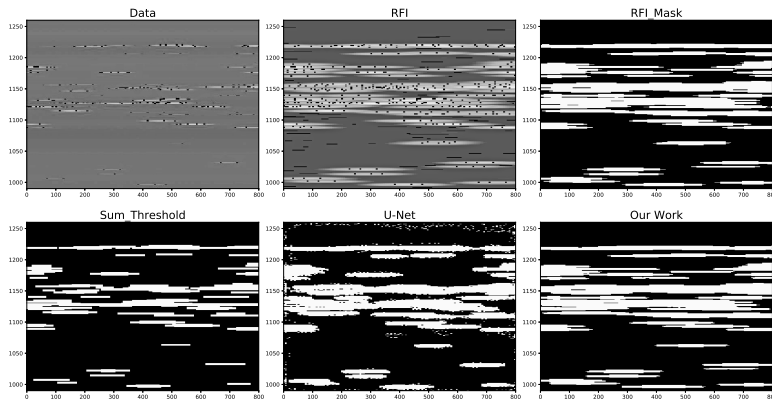


Figure 2. Comparison of experiment results of RFI recognition methods. The white zones represent recognized RFIs that is compared with accurate RFI patterns

The experiment results are shown in Fig. 2. It can be seen that the recognition result of the U-NetIM is closest to Ground_Truth (RFI_Mask), the most of RFI patterns have been correctly labeled and little misjudge occurs.

The recognition results are then further evaluated using synthetic indicators, as shown in Fig. 3. First, it can be seen that the U-NetIM achieves the highest scores for all indicators. It can not only identify more accurately, but also more completely. Second, it clearly shows that two deep learning methods (U-NetIM and U-Net model) have better performance than the traditional ones (KNN and SumThreshold) in RFI recognition. Compared to the U-Net model, U-NetIM can achieve higher recognition accuracy without sacrificing false detection rate.

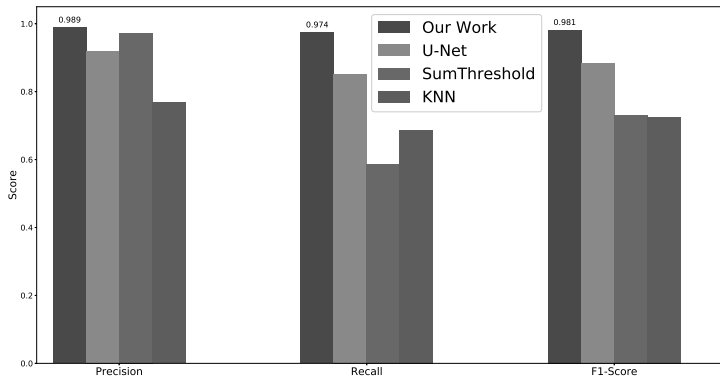


Figure 3. Evaluation of experiment results with indicators of precision, recall and F1-score. U-NetIM has the best scores in all aspects of evaluation

The better performance in deep learning methods is mainly due to a fact that the deep learning is very capable for nonlinear problems. It adopts the nonlinear activation function "ReLU", which makes it more competent for more complicated tasks. Since most form of RFIs are nonlinear, in addition, the range and size are always different, it's a challenge for the traditional methods (such as SVD) to retain good results compared to U-NetIM.

Currently, there is still room for improvement of the speed of running U-NetIM RFI recognition. In order to be able to effectively process upcoming large data set from the FAST radio telescope, the next work needs to improve the speed of recognition.

Acknowledgments. This work is supported by the Joint Research Fund in Astronomy (U1731125, U1531111) under a cooperative agreement between the National Natural Science Foundation of China (NSFC) and Chinese Academy of Sciences. M.L. thanks to the support of startup fund at Boise State University.

References

- Akeret, J., Seehars, S., Chang, C., Monstein, C., Amara, A., & Refregier, A. 2017, *Astronomy and Computing*, 18, 8
- Mosiane, O., Oozeer, N., Aniyar, A., & Bassett, B. A. 2017, in *Materials Science and Engineering Conference Series*, vol. 198, 012012
- Offringa, A. R., de Bruyn, A. G., Biehl, M., Zaroubi, S., Bernardi, G., & Pandey, V. N. 2010, *MNRAS*, 405, 155. 1002. 1957
- Ronneberger, O., Fischer, P., & Brox, T. 2015, *ArXiv e-prints*, arXiv:1505.04597. 1505. 04597

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Automatic Detection of Microlensing Events in the Galactic Bulge using Machine Learning Techniques

Selina Chu,¹ Kiri L. Wagstaff,¹ Geoffrey Bryden,¹ and Yossi Shvartzvald²

¹*JPL, Caltech, Pasadena, CA, USA; Selina.Chu@jpl.nasa.gov*

²*IPAC, California Institute of Technology, Pasadena, CA, USA*

Abstract. The Wide Field Infrared Survey Telescope (WFIRST) is a NASA flagship mission scheduled to launch in mid-2020, with more than one year of its lifetime dedicated to microlensing survey. The survey is to discover thousands of exoplanets near or beyond the snowline via their microlensing lightcurve signatures, enabling a Kepler-like statistical analysis of planets at ~1-10 AU from their host stars. Our goal is to create an automated system that has the ability to efficiently process and classify large-scale astronomical datasets that missions such as WFIRST will produce. In this paper, we discuss our framework that utilizes feature selection and parameter optimization for classification models to automatically discriminate different types of stellar variability and detect microlensing events.

1. Introduction

Microlensing is an important technique for exoplanet detection and characterization. The success of a microlensing survey relies on the amount of microlensing events detected, which is dependent on the density of observable stars. The Galactic bulge is where the stellar surface density is highest. So naturally, microlensing surveys should concentrate their efforts toward the bulge to maximize the event rate. Traditionally, these surveys have been conducted at optical wavelengths, which suffer from high dust extinction near the Galactic bulge. Observing in the near-infrared (NIR) will mitigate the effects of high extinction, enabling observations closer to the Galactic center. To understand this potential, WFIRST will conduct its microlensing survey in the NIR (1-2 μm). However, until recently there has been little or no effort dedicated to microlensing surveys in the NIR. This means we do not yet have a mapping of the microlensing event rate near the galactic center in the NIR, which makes it impossible to properly optimize WFIRST's science yield. The goal of our work is to directly address these issues by determining the optimal target fields for the WFIRST microlensing survey and developing data analysis tools to enhance the science return of the survey. We propose a framework that could efficiently process lightcurves extracted from the tens of millions of stars in an NIR survey and fully automate identification of microlensing events. In this paper, we will describe our approach in developing a predictive model using machine learning to detect microlensing events. We demonstrate our proposed method on datasets acquired from UKIRT's wide-field near-IR camera that surveys the galactic bulge.

2. Microlensing Survey

In order to study the detection of microlensing events, we started an NIR survey with the United Kingdom Infrared Telescope (UKIRT), a 3.8-m telescope on Mauna Kea in Hawaii for our investigation. UKIRT was initially started as a pilot study for the 2015 Spitzer microlensing campaign, but in 2017, the program was redirected to cover all potential WFIRST fields, including the Galactic center. The full catalog of data from the UKIRT microlensing campaigns is publicly available in the NASA Exoplanet Archive <https://exoplanetarchive.ipac.caltech.edu/docs/UKIRTMission.html>. Using the UKIRT data, Shvartzvald et al. (2017) successfully identified the first five microlensing events in the NIR based on preliminary analysis

To represent each lightcurve, we derived features using a grid-based approach for microlensing fit based on a method proposed in Kim et al. (2018). The model grid utilizes the effective event timescale t_{eff} and the event peak time t_0 , and each model lightcurve is scaled by the source flux F_s and blended flux F_b to derive analytically the best fit for each grid model. We utilize Markov Chain Monte Carlo (MCMC) techniques to approximate fits for large number of lightcurves efficiently Foreman-Mackey et al. (2013). We first select lightcurves that exhibit evidence for microlensing by comparing each fit against a straight line, using $\Delta\chi^2 \equiv (x_{flat}^2 - x_{microlens}^2)/(x_{microlens}^2/N_{points} - 4)$, where $x_{microlens}^2$ is the microlensing fit and N_{points} is the number of points in a lightcurve. For each lightcurve, we extracted 66 features; examples can be found in Table 1.

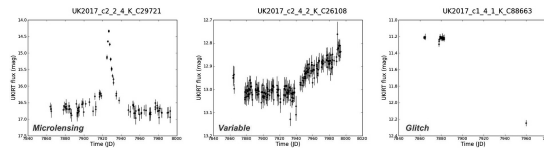


Figure 1. Examples of lightcurves: Microlensing (Left), Variable (Center), Glitch (Right)

3. Classification of Lightcurves with Feature Selection and Model Selection

For this work, we use the 2017 survey data, which was initially filtered using $\Delta\chi^2 > 100$, resulting in a subset of approximately 30,000 potential microlensing candidates. The first step in building prediction models is to obtain a set of ground truth labels to train our classifiers. To accomplish this task, we manually labeled 1,587 lightcurves by visual inspection into three event types (or classes): *microlensing*, intrinsic stellar variability events (*variables*), and spurious instrumental artifacts (*glitches*) Fig. 1. This results in 137 microlensing events, 1,083 variables, and 367 glitches. The rest of the unlabeled lightcurves are used as test candidates for detection. We evaluated three lightcurve classification methods: *Random Forest* (RF), and *Support Vector Machine* (SVM), *K-Nearest Neighbor* (kNN), along with feature selection and model selection to optimize the classifiers' parameters. When using a large number of features, there might be potentially irrelevant features that could negatively impact the quality of classification. Adding more features is not always helpful; as the number of features increases, the number of dimensions in the search space also increases, resulting in the data points becoming more sparse. We use feature selection to choose a more effective subset of features, which can reduce the overall computational cost and running time,

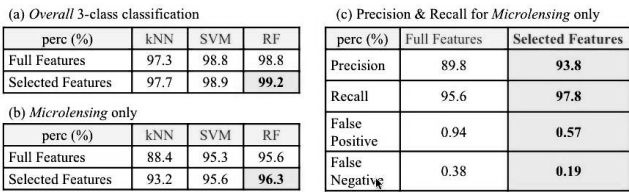


Figure 2. Summary of classification results: (a) overall; (b) and (c) microlensing class only

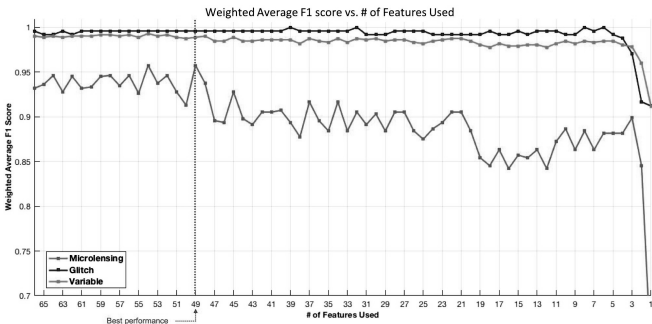


Figure 3. The effect on classification F1-score as the number of features are removed. Best performance was found using 49 features. The first ten features are list in Table 1

as well as achieve an acceptable, if not higher, recognition rate. Instead of performing an exhaustive search of all features, we use a greedy backward elimination feature selection algorithm for our experiments. Almost every classification algorithm requires some hyperparameter tuning. To optimize the performance of our algorithm, we use 3-fold cross validation to optimize the parameters for each classifier. This process was developed based on Scikit-learn scikit-learn.org and feature selection from MLxtend <http://rasbt.github.io/mlxtend>.

For our experiment, we used the filtered 2017 UKIRT survey data and trained the classifiers to differentiate between *microlensing*, *variable*, and *glitch*. We used 3-fold cross-validation for model selection and feature selection. Data are normalized using zero mean and unit variance. F1-score was used for evaluation metric. F1-score is the harmonic average of the precision and recall, defined as $F1 = 2/(recall^{-1} + precision^{-1})$. The classification results are summarized in Fig 2. Fig 3 shows the plots of the F1-score with respect to each class, as the number of features decreases. To obtain the F1-score for each class, the averaged F1-score is weighted relative to the amount of actual observations in each class. We found results using SVM and RF to be similar with RF performing slightly better overall. To demonstrate the performance of our detector, we focus on using RF. The selected model parameters Θ_s utilizes 500 number of trees in a forest, max. depth of 4 for each tree, entropy criteria, and max $\log_2(N_{features})$ features at each split. Using feature selection, the best performance was found using 49 features. Fig. 4 illustrates the lightcurves detected as microlensing event from the set of unlabeled test candidates using Θ_s with class probability (of being microlensing) $p_m(x) \geq 0.8$. We consider examples in Fig. 5 as false positives.

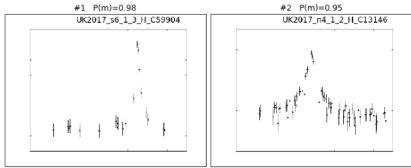


Figure 4. Examples of detected microlensing events

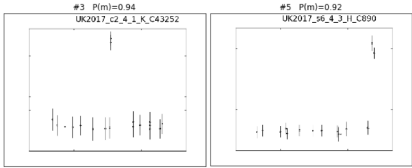


Figure 5. Examples of false positive microlensing events

Table 1. List of the top ten features found from using feature selection in Fig. 4

Rank	Feature	Description
1	$\Delta\chi^2_{drop1,drop2}$	significance level for drop-1-point vs drop-2-points
2	t_{eff}	microlensing event timescale (mcmc fit)
3	b	intercept of linear fit
4	$t_0 - t_{mid}$	how well event falls within observing window
5	m	slope of linear fit
6	$\Delta\chi^2_{linear}$	significance level for linear vs baseline fit
7	t_E	microlensing event timescale (mcmc fit)
8	F_{sin}	median flux of sinusoidal fit
9	u_0	microlensing event impact parameter (mcmc fit)
10	$N_{high,3-\sigma}$	number of points 3- σ above the baseline

4. Conclusions and Future Work

This paper investigates techniques for developing a microlensing detection based on the proposed features. The classification system was successful in classifying the different types of events. In using forward feature selection for classification, we were able to achieve a slightly higher recognition rate and improving the overall classification. Currently, we are developing a framework for injecting mock stars with microlensing signals into the UKIRT images to evaluate detection efficiency. Ongoing work includes incorporating active learning into our system to improve classification performance of lightcurves by automatically selecting most informative unlabeled lightcurves, visually labelling the selected lightcurve, and then re-training our prediction models. To assure our ability to transfer our findings from UKIRT to WFIRST effectively, we plan to incorporate domain adaptation to bridge any differences between the two surveys.

Acknowledgments. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

References

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, Publications of the Astronomical Society of the Pacific, 125, 306. 1202.3665

Kim, D.-J., Kim, H.-W., Hwang, K.-H., Albrow, M. D., Chung, S.-J., Gould, A., Han, C., Jung, Y. K., Ryu, Y.-H., Shin, I.-G., Yee, J. C., Zhu, W., Cha, S.-M., Kim, S.-L., Lee, C.-U., Lee, D.-J., Lee, Y., Park, B.-G., Pogge, R. W., & Collaboration, T. K. 2018, The Astronomical Journal, 155, 76

Shvartzvald, Y., Bryden, G., Gould, A., Henderson, C. B., Howell, S. B., & Beichman, C. 2017, The Astronomical Journal, 153, 61

Session III

Data Science: Workflows, Hardware, Software, Humanware

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Massive Data Exploration in Astronomy: What Does Cognitive Have To Do With It?

Kirk Borne

Booz Allen Hamilton, Annapolis Junction, Maryland, USA;
borne_kirk@bah.com

Abstract. There has been a tendency for astronomers to avoid unsupervised data exploration, due to the characterization of this approach as a non-scientific fishing expedition. But, a cognitive approach to massive data exploration has the potential to amplify hypothesis formulation and question generation for greater astronomical discovery. The incorporation of contextual data from other wavelengths and other surveys provides the basis for seeing interestingness in the multi-dimensional properties of sources that might otherwise appear uninteresting in a single survey catalog. Some suggested methods for cognitive exploration will be presented, including computer vision algorithms that “see” emergent patterns in the multi-dimensional parameter space of astronomical data.

1. Background: A Primer on Big Data and Astroinformatics

Where does discovery in science start? Does it start with data? Or with a hypothesis? Or with a story? Our answers to these questions will emerge below.

Ever since humans first explored our world, we have asked questions about everything that we see around us. Those questions motivate us to gather evidence (observational data) to help us answer our questions. The gathered evidence (data) usually evokes new questions, which leads to more data, etc. In an approximate sense, the amount of new data collected is proportional to the number of questions we are asking, which is proportional to the amount of data that we are examining in response to previous questions we were asking. The growth rate of data is thus approximately proportional to the amount of existing data, which is the equation of exponential growth. This is how we have ended up with exponentially growing data volumes, and this is why we now find ourselves in the “era of big data.”

Data volumes have been growing exponentially (with a doubling time of roughly two years) in nearly all areas of human experience. The growing need for knowledge discovery from these massive data collections imposes ever-increasing requirements on data services. In the sciences, the challenges are nearly insurmountable as research programs are collecting multiple terabytes of data daily, and even much more in some domains (e.g., high-energy physics). Some astronomical instruments (such as Square Kilometer Array) will generate over 100 terabytes per second. Data streams from sensors and simulations are increasingly complex in addition to growing in volume. Novel, scalable algorithms for machine learning in large-scale data offer a viable approach to the knowledge discovery challenges in many domains, and astronomy is often seen as leading the charge (Borne 2013). Astroinformatics has emerged as a new astronomy

subdiscipline in response to these big data, machine learning, data mining, and data science research and education requirements, challenges, and opportunities (Borne 2010).

Attention in this paper is given primarily to astronomical catalogs – i.e., the derived information products from raw data (such as images, spectra, photon counts, interferograms, and time series). For example, the LSST (Large Synoptic Survey Telescope) will produce 100-200 petabytes of imaging data during its 10-year “cinematographic” survey of the sky, from which a 20-petabyte scientific parameter database will be produced (Ivezić et al. 2008). The discovery potential from this catalog will be enormous, but that potential will only be achieved if sufficiently scalable machine learning algorithms are applied to the mining of those data. The corresponding data mining applications include discovery of patterns in images and other complex data types, but we will not focus on those. Instead, the focus here will be on discovery of patterns in the high-dimensional catalogs of astronomical parameters, acquired from multi-wavelength and multi-messenger instruments.

While our databases are growing in volume exponentially, a much larger “big data” challenge is the fact that multi-parameter exploration is a combinatorial problem: exploring and analyzing all possible combinations of N parameters to discover interesting patterns is an insurmountable challenge. The combinatorial growth rate of these multi-parameter combinations grows much faster than exponential: something like N^N (combinatorial) compared to 2^N (exponential). Elaborating on specific algorithms to address the combinatorial challenge is not the main focus of this paper, but we note that the set of algorithms covers link (association, graph) analysis, network science, and topological data analysis (TDA) – these algorithms attempt to mimic cognitive activity in neural systems and consequently are applied frequently in neuroscience research (Saggar et al. 2018). We often say that “*the natural data structure of the world is not rows and columns, but a graph*” – where facts are represented as nodes in the graph and knowledge as connections (edges, relationships) in the graph. We incorporate these algorithmic approaches generally in this paper under the umbrella of bio-inspired cognitive analytics (discussed in the next section). It is important to remember that graph-based algorithms are evolutionary-optimized (selected by nature) as the best approach to pattern detection and pattern recognition in our feature-rich universe.

Large time-domain surveys will be a fertile area for machine learning-based pattern and anomaly discovery. This includes not only the raw time series data, but also includes the catalogs of extracted parameters that represent the characteristic signatures (features) of detected events. For example, LSST will monitor the sky nightly for new astronomical sky events – it is anticipated that at least one million (and perhaps as many as ten million) such events will be detected each night for 10 years. The features that represent the time series (e.g., pulse shape characterization and condensed representations of the temporal patterns) must be curated in order to enable search, pattern mining, and feature learning across catalogs of many millions of sources.

Addressing the knowledge discovery challenges of this massive data mining problem is the goal of cognitive analytics, which we will describe below. The rapid detection, characterization, and analysis of interesting phenomena and of emergent behavior in high-rate data streams are often critical aspects of sky survey science, enabling data-driven decision support, instrument-steering, and prompt follow-up observation of the most interesting phenomena. Prompt follow-up observation requires accurate and meaningful characterization and interpretation of real-time events, including inte-

gration of features (derived characterizations and inferred astrophysical properties) of those sources extracted from other astronomical databases and object catalogs.

Large-scale sky survey data for static (non-variable) sources are high-rate parameter streams that describe the sources being ingested into massive multivariate data catalogs on a continuous basis. For example, a typical LSST visit consists of a new image pair being acquired every 40 seconds, with roughly 100 million sources in each visit being parameterized and stored in the survey's source catalog.

2. Cognitive Analytics: Bio-inspired Discovery

One approach to exploratory data analysis that goes beyond traditional analysis is cognitive analytics. Traditional data analysis is often a descriptive analytics approach that is directed primarily at answering pre-determined research questions through a prescribed analysis of the data. Conversely, the cognitive approach is directed at discovering new questions that are implied by the data, thus allowing the data to inform more explicitly and uniquely the most efficacious and appropriate paths of exploratory analysis. Specifically, unusually interesting and unexpected patterns in the data call out for our attention: “that’s funny!”, “what is this?”, “why is that?”. Algorithms can be deployed on databases and on data streams to perform pattern detection (unsupervised machine learning) in existing data, and then applied to future data sets for pattern recognition (supervised learning) aimed at finding more instances of those same patterns.

Cognitive analytics (i.e., question discovery in large data) emulates the highly evolved human cognitive functions of pattern detection and anomaly discovery. These activities (and their associated algorithms) include class discovery, multi-dimensional correlation discovery, surprise (novelty, outlier) discovery, real-time change (and event) detection, state transition detection, and association (link) discovery. We will simply refer to the set of these (and other use cases) as the “cognitive analytics use cases.” Computer vision (CV, sometimes referred to as machine vision) describes a set of algorithms that specifically emulates human vision in pattern detection and recognition – consequently, CV is a form of cognitive analytics. We will describe CV further below.

Databases of features (including static and variable sources) from numerous data centers, surveys, and open data collections (e.g., published in online journals) can be cross-mined through machine learning to discover the most “interesting” scientific knowledge encoded within large and high-dimensional datasets: correlations, patterns, linkages, relationships, associations, principal components, redundant and surrogate attributes, condensed representations, object classes/subclasses and their classification rules, transient events, outliers, anomalies, novelties, and surprises.

We describe below examples of algorithms that can be applied to petascale knowledge discovery and cognitive analytics (i.e., question discovery) within the context of massive (often high data rate) multivariate astronomical catalogs. We label this as “interestingness discovery” – emulating the human cognitive capacity to see something new and interesting, and then to inquire “What is this? Why is that?”

3. Computer Vision: Seeing is Discovering

Computer vision (CV) is a subset of machine learning that traditionally is directed at pattern detection and recognition in images and videos, frequently applied in robotics

and autonomous (self-driving) systems. There is no reason to limit CV applications to images in the normal sense. A projection of a high-dimensional catalog into two dimensions can be considered a “scene”, like an image. The standard set of CV algorithms can be applied to such a scene: edge detection, explanatory feature discovery and recognition, segmentation, density variations, texture characterization, motion detection, and more. The “cognitive analytics use cases” (listed earlier) can invoke these algorithms for discovery in large databases, in conjunction with traditional multivariate techniques such as PCA (Principal Component Analysis), association analysis, anomaly (outlier) detection, void detection, clustering analysis, and correlation discovery.

In the case of astronomical catalogs, two-dimensional projected “scenes” can be generated sequentially from a grand tour technique that sweeps through the various dimensions systematically (Wegman 1992). Edge detection can be used to discover sharp features and boundaries in the numerous projected parameter spaces (including fundamental planes of astronomical parameters). Segmentation can be used to detect and discover clusters and subclusters (new classes of sources). Density variations can be used to discover changes in the relative compactness of the distribution of astronomical properties of sources across various parameter spaces. Texture characterization can be used to discover hot (high-variance) populations embedded in cold (low-variance) populations, and vice versa. The “field of streams” of stripped dwarf galaxies’ tidal debris around the Milky Way is an example of the latter – we describe this case below.

4. Dimensionality Reduction and Grand Tours

The curse of dimensionality is a manifestation of combinatorial growth. Dimensionality reduction is critical in any analysis of massive data catalogs. Being able to sift through a mountain of data efficiently in order to find the key informative and explanatory features of the collection is a fundamental required capability when analyzing high-dimensional intensely multivariate parameter catalogs. Identifying the most interesting dimensions of the data is especially valuable when visualizing high-dimensional data.

The human capacity for seeing multiple dimensions is very limited: 3 or 4 dimensions are manageable; 5 or 6 dimensions are possible; but more dimensions are difficult-to-impossible to assimilate. However, the human cognitive ability to detect patterns, anomalies, changes, or other features in a large complex scene surpasses most computer algorithms for speed and effectiveness. In this case, a “scene” refers to any small- n projection of a larger- N parameter space of variables.

In data visualization, a systematic ordered parameter sweep through an ensemble of small- n projections (scenes) is often referred to as a grand tour. The grand tour allows a human viewer of the sequence to see quickly any patterns or trends or anomalies in the large- N parameter space. Even such grand tours can miss salient (explanatory) features of the data, especially when the ratio N/n is large. Consequently, a cognitive analytics approach that combines the best of both worlds (CV algorithms and human perception) will enable efficient and effective exploration of high-dimensional data.

5. Interestingness Metrics: That’s Funny

While CV algorithms are designed to emulate human perception and our highly perceptive cognitive abilities, another approach is to apply “interestingness metrics” automat-

ically to data catalogs during ingest. When faced with a database of many hundreds of attributes (e.g., a cross-matched catalog of 500+ attributes from combined sky survey catalogs), the data end-user is unlikely to know in advance which attributes are most interesting. Consequently, the user usually ends up selecting only the small handful of attributes that are most familiar to her/him. These may not be the most beneficial for discovery or for efficient database exploration.

To address this problem, various statistical measures of interestingness are useful, including information gain, entropy, GINI coefficient, covariance analysis, PCA (principal component analysis), ICA (independent component analysis), and maximal information coefficient (Reshef et al. 2011). The resulting set of metrics can be combined into a unified objective quantifiable scoring model that presents the most interesting attributes to end-users for *efficient* and *effective* data explorations: *efficient* in the “precision” sense that the selection of the most interesting attributes for query/retrieval avoids lots of useless searches and queries; and *effective* in the “recall” sense that novel discoveries (beyond known classes and expected relationships) are made possible.

In simple terms, an “interestingness” alert should elicit the desired cognitive response “that’s funny” from the data end-user. An outlier detection threshold (i.e., outlyingness) is another type of interestingness metric that gets our attention – this topic is considered separately in the next section.

Interestingness metrics set thresholds for discovery, in order to detect novel patterns and surprising behaviors within large high-dimensional data sets. For example, rule-learning algorithms (specifically decision tree rule induction) make use of the information gain metric in order to determine which feature (database attribute) contains the most “information” when assigning an accurate classification to an object. This is the one attribute among all attributes that by itself yields the best single-attribute classifier (= the top-level decision node). After testing this attribute’s information gain value, the remaining attributes are tested again for the next best information gain. The ranking of each database attribute’s power for accurately identifying and classifying the rare classes in the catalog (i.e., the attribute’s information gain in the rare class decision nodes) provides a measure of that attribute’s “interestingness.”

Interestingness metrics are essentially scoring functions applied to the data (probably during pipeline data processing and ingest). They alert the data end-user’s attention (or a cognitive analytics discovery algorithm’s attention) to the most interesting and informative features (or combinations of features) in high-dimensional data that should be investigated further.

Another example of informative feature is a latent variable. Latent (hidden) variables can be concepts that are represented by the observed data, but are not explicitly measurable (e.g., causal factors that explain the observed features; or the “sentiment” of the author of a social media posting; or the mass accretion rate onto a black hole that can only be inferred indirectly from other measurements).

The concept of latent variables also sometimes refers to an aggregation of the observed variables into a single (or a small subset of) explanatory, predictive, or descriptive variable(s). These latent variables are observable in the sense that they are not measurable but can be imputed directly through some mathematical combination of the measured variables (i.e., through a computed function of the input variables) – this is also an explicit application of *dimensionality reduction* in large high-dimensional data. A simple example should illustrate this point: we can measure the apparent radius and magnitude for galaxies in a large sample, but these measurements are not intrinsic prop-

erties (thus, not particularly significant on their own). However, combinations of these parameters can deliver to the data end-user a more natural, intrinsic, and meaningful physical parameter (e.g., total flux divided by apparent radius for a self-gravitating system is a surrogate for mass divided by radius, which is proportional to the depth of the object's gravitational potential, which has much more astrophysical significance).

Interestingness metrics go beyond domain-specific (astrophysical) concepts to include statistical estimates of interestingness (such as those mentioned earlier). These can tell us what is interesting (unusual, or unexpected) in the data stream. When the scoring function that is used in the metric is explicitly known, then the metric can provide a deeper intuitive understanding of why a feature was flagged as interesting.

The scoring models for interestingness metrics should be transparent, perhaps adjustable, and parsimonious for the user. That is, they should result in a manageable (non-redundant) set of interpretable and insightful “alerts” derived from the data. These may include new subclasses of a known class of objects, unexpected correlations, fundamental planes (i.e., a surprisingly small number of significant principal components), outliers, voids and gaps in the data distribution, transients, changes in the stationarity of a data distribution including time series (to be described below), or data quality problems (including instrument defects, processing artifacts, or measurement errors).

Interestingness metrics that are primarily statistical in origin can still carry astrophysical significance, since there is usually some law of physics or astrophysical process that governs the observed unusual statistical behavior relative to other objects in the data catalog. In this sense, such metrics are essentially comparative metrics that signal to a cognitive algorithm (or human) to “focus your attention here at this partition in the data catalog.” For the sake of utility, the partition should be a low-dimension projection of the higher-dimension catalog.

6. Outlyingness for Surprise Discovery

In many cases, the first test for outliers is a simple distance-based statistic (how many standard deviations from the mean is the point?). Outlyingness in multivariate data will sometimes use the Mahalanobis distance measure.

A multivariate distance-based technique that is a bit surprising is one that starts with PCA. Here is an example of how it can be applied: suppose that the first two principal components (PC) capture and explain most of the sample variance. This means that the data lie approximately in a hyperplane that defines the distribution of the overwhelming majority of data points. Consequently, the third PC measures the distance of a data point from that fundamental hyperplane (Dutta et al. 2007). Simply sorting the catalog records by their coefficient on the third PC, and then identifying those with the largest values, produces a rank-ordered list of outliers for further investigation. This same technique can be applied to higher dimensions and to a greater number of principal components.

Moving beyond distance-based estimators, we developed a density-based multivariate outlier detection algorithm, which we call KNN-DD (K-Nearest Neighbors Data Distribution). The *multivariate* KNN-DD algorithm takes advantage of robust well established *univariate* statistical tests. In particular, KNN-DD samples the statistical distribution of nearest neighbor distances for a test data point (possible outlier) and compares that distribution with the statistical distribution of the distances among the set of its K nearest neighbors (Borne & Vedachalam 2012). For example, if we have a

catalog of 10000 data items, we search for the $K=100$ nearest neighbors in proximity to the test point. The distribution of 100 nearest neighbor distances (between the test point and the 100 nearest data items) is calculated. Then the distribution of $4950 = K(K-1)/2$ distances among the $K=100$ “normal” data items is calculated. These two distributions are **univariate functions of the distance**. A simple two-sample distribution comparison test (such as the K-S test) is used to test the Null Hypothesis that the two distributions are drawn from the same parent population.

The K-S (Kolmogorov-Smirnov) test is a classic non-parametric two-sample statistical test used to estimate the likelihood that two sample distributions are drawn from the same population (which is the Null Hypothesis). There is no assumption regarding the functional form of the distance distribution functions – this is an important and essential criterion in order to avoid introducing any bias in the estimation of outlier probability. Our algorithm makes no assumption about the shape of the data distribution or about “normal” behavior (hence, it is non-parametric). It simply searches for unusually (abnormally) distributed data points.

KNN-DD is applicable to high-dimensional data catalogs since it is algorithmically univariate, by estimating a function that is based entirely on the scalar distance between data points (which themselves occupy high-dimensional parameter space). KNN-DD actually compares the cumulative distributions of the test data’s inter-point distances, without regard to the nature of those distributions – therefore, the algorithm can be applied to multivariate data and yet remain a univariate test, thus solving the curse of dimensionality. The algorithm’s distance metric is automatically extensible to higher dimensions; and the distance distributions are computed only on small- K local subsamples of the full dataset of N data points ($K \ll N$). Consequently, the KNN-DD density-based outlier detection algorithm is easily (embarrassingly) parallelizable when testing multiple data points for outlyingness.

A more general approach to outlier detection goes beyond the specificity of distance and density estimators. That would be a pattern-based approach that can be generalized to any pattern or anomaly (surprise, novelty) in the data catalog. A handy hierarchical classification taxonomy for outlier detection algorithms therefore incorporates these three different approaches:

- Outlier detection 1.0 = distance-based (where the point of interest is at a relatively large distance from the mean value of typical data points). This is a traditional statistical method. Detection thresholds can be set to 3, 4, 5, or 6 standard deviations (or more) from the mean, depending on the desired (manageable) size of the resulting list of potential outliers (surprises) to be derived from the data set.
- Outlier detection 2.0 = density-based (where the local density at the point of interest is different from the local density around typical data points, either significantly higher or lower – this includes points of interest that are deeply embedded within the data cloud, hence not flagged by a distance-based approach). This is a kernel-based method. The larger the data set, then the more statistically robust is the kernel at approximating the real data distribution.
- Outlier detection 3.0 = pattern-based (where the pattern of a set of data points in parameter space is atypical, including unusual trends, correlations, principal components, clusters, subclasses, associations, asymmetric distributions, etc.). This is a cognitive method that elicits the desired cognitive reaction from such an unusual discovery: “that pattern looks funny.”

7. Sniffing Out Cold Cases with DOGs

Computer vision (CV) algorithms include edge-detection, gradient-detection, motion-detection, change-detection, segmentation, template-matching, and pattern recognition. As stated earlier, many of these can be applied to high-dimensional data catalogs, not simply to 2-dimensional images. As an example, we consider a method that has been used to discover tidal stellar streams around the Milky Way – the “field of streams” (Belokurov et al. 2006) – the remnants of tidally shredded dwarf galaxies that were the hierarchical building blocks of mass assembly that has produced our home galaxy. The old stars that comprise these streams are very sparsely distributed and spread widely across the whole sky, and yet they are actually discoverable and distinguishable against the rich background of Milky Way stars. This is because the stellar distributions within the tidal streams are “cold” – dynamically cold (low velocity dispersion across the stream), spatially cold (low spatial cross-section of the extremely long narrow stream), and photometrically “cold” (low dispersion in colors [due to their similar age and metallicity] relative to the more diverse populations in a random Milky Way star field).

Searching for tidal streams is therefore a great application of CV – finding edges, narrow features, and sharp (cold) patterns against a relatively smooth stellar background that has high dispersion in velocity, spatial extent, and color. It is precisely this distinction between the cold (low variance) and hot (high variance) components in the data distribution that enables CV to discover interesting (though very rare) features in the data. Applying this CV approach more generally to large catalog hyper-parameter space explorations could potentially yield new and unexpected discoveries. The CV algorithm that has worked well in the discovery of the “field of streams” around the Milky Way is the DOGs method (Koposov 2008). “DOGs” refers to the *Difference Of Gaussians* (also known as the Marr-Hildreth algorithm, or high-pass filter):

$$S(x, y) = I(x, y) * G(x, y, \sigma_1) - I(x, y) * G(x, y, \sigma_2), \text{ with } \sigma_2 \gg \sigma_1$$

Here, $I * G$ specifies the convolution of the observed density field $I(x, y)$ with a kernel density function (a Gaussian) $G(x, y, \sigma)$, where σ is the width (standard deviation) of the Gaussian distribution, and (x, y) can be any arbitrary 2-D projection of a multivariate parameter space (perhaps a “scene” from a grand tour). The result of this subtraction of the smooth component from the data distribution is the “appearance” of narrow, sharp, cold features in the “scene” (due to the application of the high-pass filter).

This cognitive analytics use case can be applied in general parameter spaces, in spatial dimensions (x, y) , and in the time domain to track down cases of cold (astrophysically confined) features in large sky survey catalogs. Discoveries may include unexpected confinement (e.g., highly flattened ellipsoidal distributions) of astrophysical parameters in some parameter spaces (akin to the field of streams, or the fundamental plane of elliptical galaxies), or unusually dense subclasses of known classes of objects, or “cold” signals (e.g., harmonic ringing) in time series emerging from a background of “hot” white noise (e.g., LIGO precursors in multi-messenger surveys).

8. Eigenstate Drift Detection through Condensed Representations

Condensed representations are a particularly useful dimensionality reduction approach that enables cognitive discovery of changes in stationarity of a population across any parameter space. A condensed representation of a data stream refers to a small set of coefficients (e.g., eigenvalues of the principal components; Fourier coefficients; or

wavelet representations) that compactly describe a much larger data payload to another algorithm (e.g., a cognitive analytics use case). Though this is most useful in the context of mining temporal data streams, nevertheless eigenstate drift detection (i.e., discovery of non-stationarity) is a general application in which a compact parametric characterization and exploration of pattern changes in a large data set can be conducted across any independent variable in parameter space.

Changes in the stationarity of a population across the sky, or across different density regimes within a galaxy cluster or star-forming region, or across different values of any preferred astrophysical parameter can be discovered through changes in the eigenstates (including eigenvectors and eigenvalues) of the data distribution. Because these eigenstates provide convenient and efficient short-hand (condensed) representations of the features contained in the full data payload, eigenstate drift detection can be explored to address countless questions related to the invariance of astrophysical phenomena in different environments across different physical conditions.

As an example of eigenstate drift detection, we analyzed the principal components of galaxy parameters as a function of an independent variable, similar to temporal stream mining, which would have time as the independent variable. For our experiments, the independent variable was the galaxy's local galaxy density (i.e., the number density of nearby galaxies within its spatial environment). The class of elliptical galaxies has been known for more than 25 years to show dimensionality reduction among a subset of their observed properties, such that the 3-dimensional distribution of three of those astrophysical parameters reduces to a 2-dimensional hyperplane (referred to as the fundamental plane). The direction of the normal vector to that plane is a condensed representation of that plane. We investigated changes of the fundamental plane as a function of local galaxy density by measuring the direction of this vector within 30 density bins (Das et al. 2009). Computing these vectors for a large number of galaxies, one density bin at a time, measures the stationarity of galaxy properties across different density regimes in the Universe. Our research uncovered these cosmological results: we find that the variance captured in the first two principal components increases systematically from low-density regions to high-density regions (the plane gets "colder"), and the direction of the plane's normal vector also drifts systematically in the 3-D parameter space when it is derived serially from the lowest density regions up to the highest density regions (Bhaduri et al. 2011).

Parameter sweeps of other high-dimensional catalogs across other independent variables may discover similar state (phase) transitions, onset of change points, non-stationarity, and pattern drift in correlations that are presumed to be isotropic, homogeneous, or invariant across different astrophysical environments.

9. Summary: The Powers of Three

Our 3-step cognitive approach to exploration and discovery in astronomically (and combinatorially) growing astronomy data collections follows these *three C's*:

1. *Characterize* and capture forensic features from the data at large scale through learned patterns and trained algorithms, using various approaches: (1) machine-identified; (2) scientist-identified; and (3) crowdsourced (e.g., through Citizen Science, tapping into the power of natural human cognition on a global scale to find patterns and anomalies in massive data collections). That's cool.

2. *Contextualize* the data features with rich annotations: the astronomical context, the time, the scientific use cases, the extracted results, the user communities, the instrument, etc. = capture the where, when, why, what, who, and how. That's Metadata! Metadata! Metadata! That's captured content and context.
3. *Curate* these data features for search, re-use, exploration, and new question-generation (and include links to related parameters and associated features from other data sources and catalogs) = That's cognitive discovery!

Therefore, our answers to the three questions that we presented at the beginning are:

- Does discovery in science start with data? *Yes*, because observations of the world around us are what trigger our inquisitiveness and curiosity!
- Does discovery start with a hypothesis? *Not really*, because the hypothesis starts as an inference from data (observation).
- Does discovery start with a story? *Maybe yes*, because the story is inspired by data and the story leads us to seek explanations and answers to the questions that the data evoke within us.

References

- Belokurov, V., Zucker, D. B., Evans, N. W., Gilmore, G., Vidrih, S., Bramich, D. M., Newberg, H. J., Wyse, R. F. G., Irwin, M. J., Fellhauer, M., Hewett, P. C., Walton, N. A., Wilkinson, M. I., Cole, N., Yanny, B., Rockosi, C. M., Beers, T. C., Bell, E. F., Brinkmann, J., Ivezić, Ž., & Lupton, R. 2006, *ApJ*, 642, L137. [astro-ph/0605025](https://doi.org/10.1086/5025)
- Bhaduri, K., Das, K., Borne, K. D., Giannella, C., Mahule, T., & Kargupta, H. 2011, *Statistical Analysis and Data Mining*, 4, 336. URL <https://doi.org/10.1002/sam.10120>
- Borne, K. 2013, in *Planets, Stars and Stellar Systems. Volume 2: Astronomical Techniques, Software and Data*, edited by T. D. Oswalt, & H. E. Bond, 403
- Borne, K. D. 2010, *Earth Science Informatics*, 3, 5. URL <https://doi.org/10.1007/s12145-010-0055-2>
- Borne, K. D., & Vedachalam, A. 2012, in *Statistical Challenges in Modern Astronomy V*, edited by E. D. Feigelson, & G. J. Babu, vol. 902, 275
- Das, K., Bhaduri, K., Arora, S., Griffin, W., Borne, K. D., Giannella, C., & Kargupta, H. 2009, in *Proceedings of the SIAM International Conference on Data Mining, SDM 2009*, Sparks, NV, USA, 247. URL <https://doi.org/10.1137/1.9781611972795.22>
- Dutta, H., Giannella, C., Borne, K. D., & Kargupta, H. 2007, in *Proceedings of the SIAM International Conference on Data Mining, SDM 2007*, Minneapolis, MN, USA, 473. URL <https://doi.org/10.1137/1.9781611972771.47>
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., Allsman, R., Alonso, D., AlSayyad, Y., Anderson, S. F., Andrew, J., & et al. 2008, *ArXiv e-prints*. 0805.2366
- Koposov, S. 2008, in *American Institute of Physics Conference Series*, edited by C. A. L. Bailer-Jones, vol. 1082 of *American Institute of Physics Conference Series*, 233
- Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., & Sabeti, P. C. 2011, *Science*, 334, 1518
- Saggar, M., Sporns, O., Gonzalez-Castillo, J., Bandettini, P. A., Carlsson, G., Glover, G., & Reiss, A. L. 2018, *Nature Communications*, 9, 1399
- Wegman, E. J. 1992, in *Computing Science and Statistics*, edited by C. Page, & R. LePage (New York, NY: Springer New York), 127

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

A New Synthesis Imaging Tool for ALMA Based on Sparse Modeling

Takeshi Nakazato,¹ Shiro Ikeda,² Kazunori Akiyama,^{1,3,4} George Kosugi,¹
 Masayuki Yamaguchi,^{1,5} and Mareki Honma¹

¹*National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo, Japan; takeshi.nakazato@nao.ac.jp*

²*The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo, Japan*

³*National Radio Astronomy Observatory, 520 Edgemont Rd, Charlottesville, VA, USA*

⁴*Massachusetts Institute of Technology, Haystack Observatory, 99 Millstone Rd, Westford, MA, USA*

⁵*Department of Astronomy, Graduate School of Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan*

Abstract. A new imaging tool for radio interferometry has been developed based on the sparse modeling approach. It has been implemented as a Python module operating on Common Astronomy Software Applications (CASA) so that the tool is able to process the data taken by Atacama Large Millimeter/submillimeter Array (ALMA). In order to handle large data of ALMA, the Fast Fourier Transform has been implemented with gridding process. This imaging tool runs even on a standard laptop PC and processes ALMA data within a reasonable time. The interface of the tool is comprehensible to CASA users and the usage is so simple that it consists of mainly three steps to obtain the result: configuration, gridding, and imaging. A remarkable feature of the tool is that it produces the solution without human intervention. Furthermore, the solution is robust in the sense that it is less affected by the processing parameters. For the verification of the imaging tool, we have tested it with two extreme examples from ALMA Science Verification Data: the protoplanetary disk, HL Tau as a typical smooth and filled image, and the lensed galaxy, SDP.81 as a sparse image. The comparison between our results and those of traditional CLEAN method will also be provided.

1. Introduction

In radio interferometry, synthesis imaging from the observed visibility is a crucial step to investigate the detailed spatial structure of the astronomical source. Although the visibility is basically a Fourier Transform of the intensity distribution on the sky, the imaging process is not that simple. An essential and unavoidable problem is that visibility measurement is incomplete so that only part of the Fourier components is known. The most successful algorithm to overcome this is the CLEAN algorithm, which mainly

consists of two steps: Fourier Transform to obtain the image with artifacts due to incomplete visibility sampling, and deconvolution to remove those artifacts.

On the other hand, different approaches have recently been introduced to interferometric imaging. We employ the sparse modeling approach, which is known as a technique to solve an underdetermined problem. The method has already been applied to the imaging of Very Long Baseline Interferometry (VLBI) data and it has been shown that the sparse modeling technique is promising method for synthesis imaging (Honma et al. 2014; Akiyama et al. 2017). Based on these results, we have developed the Python module for Radio Interferometry Imaging with Sparse Modeling (PRIISM) to perform synthesis imaging on the data taken by Atacama Large Millimeter/submillimeter Array (ALMA). To facilitate the processing of ALMA data, PRIISM operates on Common Astronomy Software Applications (CASA) and accepts visibility data as a native data format of CASA. In this paper, we briefly describe underlying mathematics and basic usage of PRIISM. We also demonstrate the capability of PRIISM based on the ALMA Science Verification data and comparison with those images generated by CLEAN.

2. Formulation

We formulate the synthesis imaging as least square minimization problem with penalty terms. Given observed visibility v , we find the image x that minimizes

$$\frac{1}{2}|v - F(x)|^2 + \lambda_1|x| + \lambda_{\text{TSV}}\text{TSV}(x), \quad (1)$$

where λ_1 and λ_{TSV} are regularization parameters for two penalty terms. In equation 1, F represents Fourier Transform operator and TSV is the Total Square Variation (TSV) of the intensity distribution (Kuramochi et al. 2018), which is a squared sum of difference between neighboring data. For two-dimensional data, it is represented as

$$\text{TSV}(x) = \sum_{i,j} \{(x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2\}. \quad (2)$$

The first term in equation (1) is an ordinary least-square term while the second and third terms are the penalty terms, which are the indicator of a sparsity and a smoothness of the solution, respectively. In each pair of λ_1 and λ_{TSV} , we obtain one image as a solution of the minimization problem. The k -fold cross validation is used to determine the best choice of λ_1 and λ_{TSV} (The value of k can be customized. By default, the 10-fold cross validation is performed). The choice of these two parameters strongly depends on the property of the observed source.

3. Usage of PRIISM

PRIISM is a Python implementation of synthesis imaging problem described in section 2. It has a comprehensible API specific for ALMA (`priism.alma`) as well as a primitive, extendable API, `priism.core`. Typical usage of the `priism.alma` consists of three steps: (1) configuration including solver setup, data selection, image configuration, and gridding setup, (2) visibility gridding, and (3) solve the problem and find the best image by means of the cross-validation. Since PRIISM accepts visibility data as

a native data format of CASA, any ALMA data imported into CASA can directly be processed with PRIISM. Example script will be provided with the source code, which only consists of less than 60 lines including comments and spaces for readability. Users of PRIISM can use the example script as a template to process their visibility data. This will facilitate the imaging with PRIISM. With user-specified parameter ranges for λ_1 and λ_{TSV} , PRIISM generates the resulting image automatically. The parameter range should be refined until it hits the most reliable pair of parameters. To accomplish this, two or more iterations might be required. Installation of PRIISM is easily done through `cmake`. Once all the prerequisites are satisfied, PRIISM will immediately be available by running `cmake` with appropriate options followed by `make install`. Currently, PRIISM only supports 1-channel continuum imaging of Stokes I although it accepts multi-channel, multi-polarization (correlation) visibility data. Spectral line and full Stokes imaging will be implemented in the future.

4. Application to the ALMA Science Verification Data

As a verification and demonstration of PRIISM, we generated images from ALMA Science Verification data¹. We selected two images from the 2014 ALMA Long Baseline Campaign: the protoplanetary disk, HL Tau (ALMA Partnership et al. 2015a), and the lensed galaxy, SDP.81 (ALMA Partnership et al. 2015b). Kosugi et al. (2019) found that the solution immediately converges with modest number of iterations in the case of ALMA data. Therefore, we set maximum number of iterations to 1000 to reduce the computational cost. Indeed, the difference between the images obtained by 1000 and 10000 iterations was negligible. We compared our results with the images provided as a reference that are generated by CLEAN. Figure 1 shows the images for HL Tau and SDP.81 obtained by PRIISM as well as the ones generated by CLEAN. As shown in the figure, PRIISM successfully reproduced striking features for both HL Tau and SDP.81. Several gap features in the disk of HL Tau can easily be recognized. Also, clumpy structure in the Einstein ring of SDP.81 coincides well with the CLEAN image. Figure 1 demonstrates that PRIISM has sufficient ability and flexibility of handling various types of images ranging from smooth, extended sources to clumpy, sparse ones. The figure also illustrates that PRIISM is an effective tool for ALMA that targets a wide variety of astronomical sources.

5. Summary

We developed a new synthesis imaging tool, PRIISM, based on the sparse modeling technique. PRIISM has an interface specific for ALMA and accepts visibility data as a native data format for CASA so that it is possible for users to process ALMA data directly. With minimal set of configurations, PRIISM produces the optimal image automatically. We compared our results with the reference images of the ALMA Science Verification data to demonstrate the capability of PRIISM. We showed that PRIISM successfully reproduced striking features of the target sources. Our result indicates that PRIISM and the underlying sparse modeling technique is a promising tool for the synthesis imaging.

¹<https://almascience.nao.ac.jp/alma-data/science-verification>

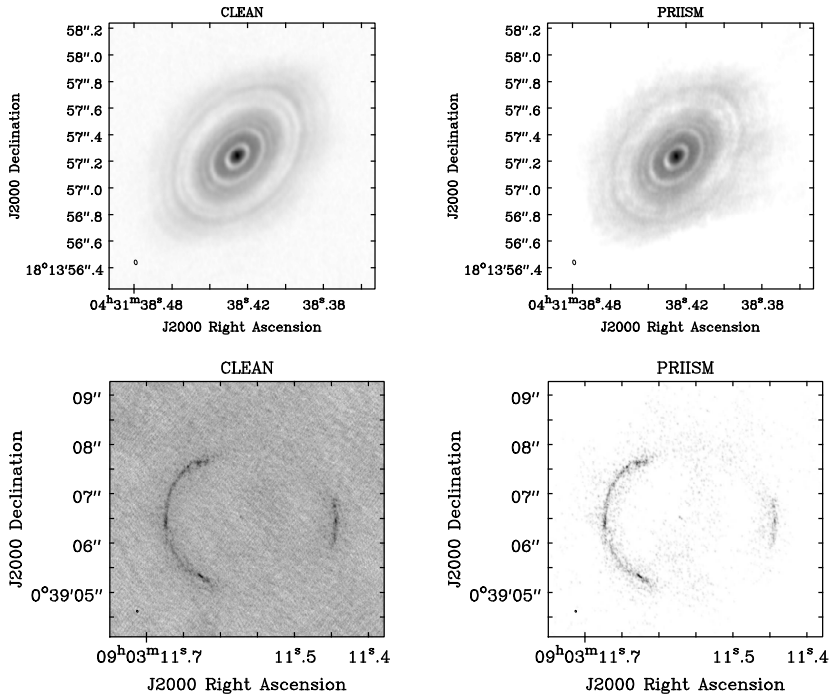


Figure 1. *Top Left:* HL Tau image generated by CLEAN (ALMA Partnership et al. 2015a). *Top Right:* HL Tau image generated by PRIISM. *Bottom Left:* SDP.81 image generated by CLEAN (ALMA Partnership et al. 2015b). *Bottom Right:* SDP.81 image generated by PRIISM.

Acknowledgments. This paper makes use of the following ALMA data: ADS/JAO.ALMA#2011.0.00015.SV and ADS/JAO.ALMA#2011.0.00016.SV. ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), MOST and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ.

References

- Akiyama, K., et al. 2017, *ApJ*, 838, 1
 ALMA Partnership, et al. 2015a, *ApJ*, 808, L3
 — 2015b, *ApJ*, 808, L4
 Honma, M., Akiyama, K., Uemura, M., & Ikeda, S. 2014, *Publications of the Astronomical Society of Japan*, 66, 95
 Kosugi, G., Nakazato, T., & Ikeda, S. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 575
 Kuramochi, K., Akiyama, K., Ikeda, S., Tazaki, F., Fish, V. L., Pu, H.-Y., Asada, K., & Honma, M. 2018, *ApJ*, 858, 56

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Towards New Solutions for Scientific Computing: The Case of Julia

Maurizio Tomasi¹ and Mosé Giordano²

¹*Università degli Studi, Milano, Italy; maurizio.tomasi@unimi.it*

²*Università del Salento, Lecce, Italy; mose.giordano@le.infn.it*

Abstract. This year marks the consolidation of Julia (<https://julialang.org/>), a programming language designed for scientific computing, as the first stable version (1.0) has been released, in August 2018. Among its main features, expressiveness and high execution speeds are the most prominent: the performance of Julia code is similar to statically compiled languages, yet Julia provides a nice interactive shell and fully supports Jupyter. Moreover, it can transparently call external codes written in C, Fortran, and even Python and R without the need of wrappers. The usage of Julia in the astronomical community is growing, and a GitHub organization named JuliaAstro takes care of coordinating the development of packages. In this paper we present the features and shortcomings of this language, and discuss its application in astronomy and astrophysics.

1. Introduction

Julia (Bezanson et al. 2017) is a programming language that has recently reached its first stable milestone: version 1.0 has been released in August 2018, and the language specification has been frozen. Julia provides a number of features that makes it extremely interesting for astrophysics, astronomy, and scientific applications in general:

- Nice and simple syntax, similar to Matlab's.
- The speed of Julia codes often matches the speed of other languages used for High Performance Computing (HPC), namely C, C++, and Fortran, thanks to a number of features: type inference, Just-In-Time compilation, use of LLVM to produce optimized machine code;
- Native support for vectors, matrices, and tensors;
- Support for missing values (using the keyword `missing`), useful when dealing with data acquired using real-world experiments;
- First-class support for many numeric types, apart from integers and floating-point numbers: rationals, complex numbers, arbitrary-precision numbers.
- Symbolic computation (e.g., estimation of analytical derivatives) is easy to implement;
- Easy to call functions defined in dynamic libraries, using the `ccall` function;
- Ability to import packages written in Python or R; several wrappers to well-known Python libraries are available (e.g., `PyPlot.jl` wraps `Matplotlib`).

2. Features of Julia

2.1. Compilation model

Julia compiles functions the first time they are executed. The compilation depends on the type of the function parameters, as shown in this example:

```
f(x) = 2x + 1  # Define a function
f(1)           # Compile f assuming an integer argument
f(1.0)         # Compile again f assuming a float argument
f(3)           # No compilation is necessary, as 3 is an int
```

2.2. Operations on arrays, matrices, and tensors

Julia's arrays are similar to Fortran's:

1. Indices start from 1;
2. Arrays are stored in column-major order;
3. The compiler is able to propagate operators and functions to arrays, performing loop fusion.

The latter point is particularly important. If `a`, `b`, `c`, and `result` are arrays of the same size, the statement `result = a + b + c` in Fortran corresponds to one do loop. On the other side, the same code in Python applied on NumPy arrays is equivalent to the application of *three* for-loop cycles, because NumPy is not able to perform¹ *loop fusion*, i.e., the combination of several for loops into one.

Loop fusion is an important feature for HPC languages. Julia provides loop fusion through the so-called *dotted operators*: if `#` is a two-argument operator, `.#` applies the operator to all the elements of the two arrays. Therefore, in Julia the code `result .= a .+ b .+ c` is equivalent to the Fortran code `result = a + b + c`. Julia's approach is more general, as this applies to custom operators and functions as well:

```
++(a::Real, b::Real) = 2a + b      # Custom operator
3 ++ 4                             # Result: 10
[3, 4] .++ [4, 7]                  # Result: [10, 15]
f(x::Real) = 3x^2                  # Custom function
f.([3, 6, 5])                      # Result: [27, 108, 75]
```

2.3. Homoiconicity

Julia provides the syntax for manipulating its own code with the same syntax used to manipulate variables. This feature, called *homoiconicity* ("same representation"), is inspired by LISP-like languages, and it has several applications in the domain of symbolic analysis (e.g., automatic computation of analytical derivatives). An interesting applications of homoiconicity in Julia is provided by the Zygote package (Innes 2018), which is able to perform automatic symbolic differentiation at compile time:

¹This limitation can be circumvented by other libraries, like WeldNumPy (<https://www.weld.rs/weldnumpy/>), Numba (<https://numba.pydata.org/>), or Cython (<https://cython.org/>).


```
julia> using Zygote
julia> f(x) = 2x + 1
julia> @code_llvm f'(0)
; Function #68
; Location: /somewhere/interface.jl:49
define i64 @"julia_#68_37159"(i64) {
top:
ret i64 2 # Return 2 immediately (the derivative is a constant)
}
```

3. Julia in Astronomy

3.1. JuliaAstro

The JuliaAstro GitHub organization (<https://github.com/JuliaAstro>) collects all the packages related to astronomy developed for Julia. At the time of writing (November 2018), the packages are the following:

- `AstroImages.jl`: Visualization of astronomical images;
- `AstroLib.jl`: Bundle of small astronomical and astrophysical routines;
- `AstroTime.jl`: Astronomical time keeping;
- `Cosmology.jl`: Library of cosmological functions;
- `DustExtinction.jl`: Models for the interstellar extinction due to dust;
- `ERFA.jl`: Wrapper to `liberfa`²;
- `EarthOrientation.jl`: Earth orientation parameters from IERS tables;
- `FITSIO.jl`: Flexible Image Transport System (FITS) file support;
- `LombScargle.jl`: Compute Lomb-Scargle periodogram;
- `SPIICE.jl`: Julia wrapper for NASA NAIF's SPIICE toolkit;
- `SkyCoords.jl`: Support for astronomical coordinate systems;
- `UnitfulAstro.jl`: An extension of `Unitful.jl` (a package to attach measure units to variables) for astronomers;
- `WCS.jl`: Astronomical World Coordinate Systems library.

3.2. Simulating a CMB space mission

One of us (MT) has had the opportunity to use Julia in a few studies involving the design of a CMB space mission (CORE, PICO, and LiteBIRD). These studies involved the simulation of the operations needed to observe the sky, and they required the generation of simulated noisy data timelines acquired by instruments mounted onboard the spacecraft. The quantity of data was of the order of hundreds of GB, and the exploratory nature of the study made existing codes (developed in C++ for the Planck experiment) cumbersome to use, as they were conceived as large monolithic programs meant to be ran end-to-end. A rewrite of some modules in Julia provided similar performance (within 10%) with the existing C++ codes; moreover, the Julia codes were runnable in Jupyter notebooks, thus allowing to interactively explore the parameter space and ease data analysis.

²<https://github.com/liberfa/erfa>. This is a BSD-licensed replica of the SOFA library (<http://www.iausofa.org/>).

4. Conclusions

Julia has several features that make it an interesting solution for astronomical and astrophysical projects. It can achieve performance similar to compiled languages, like C and Fortran, but it is considerably more expressive and easy to use.

Notwithstanding the long list of interesting features, we believe it would not be fair to omit some of Julia's most important shortcomings:

- Compilation times can be significant. Since compilation happens at runtime, a Julia script that calls several short functions can be noticeably slower than a similar script written in other compiled or interpreted languages.
- The language is new, and there are not as many libraries as for other languages. Python, R, C, and Fortran library are easy to import; however, if a code heavily relies only on a few libraries, it is usually easier to just use the language for which these libraries were developed than wrapping everything in Julia.
- It is still not possible to produce stand-alone executables. This makes code deployment more difficult.
- As any new language, it is necessary to grasp a number of concepts before being fully productive with it. For instance, a programmer experienced in NumPy might find surprising that explicit for loop can be more performant than expressions involving broadcasting. (The repository <https://github.com/ziotom78/python-julia-c-> provides an example.)

In the opinion of the authors, there are two contexts in astrophysical data analysis where Julia can provide a significant advantage over existing solutions:

- Analysis of large amounts of data, where no existing codes are available and the amount of calculations is significant. In this case, Julia codes can be as performant as other codes written using multiple libraries and languages: the typical case uses Python for most of the code and some optimized library (Numba, Fortran codes wrapped using `f2py`) for the most performance-critical routines. As an application of this use case we mention the Celeste project, which was able to load and process 178 TB of data from the SDSS catalogue in 14.6 minutes across 8192 nodes (Regier et al. 2018).
- Existing codes are monolithic and difficult to use interactively, and the expense of rewriting code in Julia can be rewarded by the possibility to run the code interactively, either in Julia's command line or in Jupyter notebooks.

Acknowledgments. We thank the Julia community at for many useful discussions, <https://discourse.julialang.org/>

References

- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. 2017, SIAM Review, 59, 65
Innes, M. 2018, ArXiv e-prints. 1810.07951
Regier, J., Pamnany, K., Fischer, K., et al. 2018, ArXiv e-prints. 1801.10277

Performance Analysis of the SO/PHI Software Framework for On-board Data Reduction

K. Albert,¹ J. Hirzberger,¹ D. Busse,¹ J. Blanco Rodríguez,² J. S. Castellanos Durán¹, J. P. Cobos Carrascosa,³ B. Fiethe,⁴ A. Gandorfer,¹ Y. Guan,⁴ M. Kolleck,¹ A. Lagg,¹ T. Lange,⁴ H. Michalik,⁴ S. K. Solanki,¹ J. C. del Toro Iniesta,³ and J. Woch¹

¹*Max Planck Institute for Solar System Research, Göttingen, Germany*
albert@mps.mpg.de

²*Universidad de Valencia, Paterna (Valencia), Spain*

³*Instituto de Astrofísica de Andalucía (IAA - CSIC), Granada, Spain*

⁴*Institute of Computer and Network Engineering, TU Braunschweig, Germany*

Abstract. The Polarimetric and Helioseismic Imager (PHI) is the first deep-space solar spectropolarimeter, on-board the Solar Orbiter (SO) space mission. It faces: stringent requirements on science data accuracy, a dynamic environment, and severe limitations on telemetry volume. SO/PHI overcomes these restrictions through on-board instrument calibration and science data reduction, using dedicated firmware in FPGAs. This contribution analyses the accuracy of a data processing pipeline by comparing the results obtained with SO/PHI hardware to a reference from a ground computer. The results show that for the analyzed pipeline the error introduced by the firmware implementation is well below the requirements of SO/PHI.

1. Introduction

The Polarimetric and Helioseismic Imager (PHI) is one of ten instruments to orbit the Sun on-board Solar Orbiter (SO; see Müller et al. 2013). SO/PHI (Solanki et al. 2018), is an imaging spectropolarimeter, probing the photospheric Fe I 6173 Å absorption line.

SO/PHI records data in five dimensions: time series of data sets containing 2048×2048 pixel images of the Sun, sampling the target absorption line at six wavelengths, recording four different polarization states at each wavelength. These polarization states contain linear combinations of the Stokes parameters ($\mathbf{S} = [I, Q, U, V]^T$), a formalism to describe the polarization of light in terms of four ideal polarization filters. To arrive to the Stokes images (the input for scientific analysis), the recorded polarization states are demodulated with the demodulation matrix. These images, complemented with a wavelength dimension, encode the magnetic field vector at the mean formation height of the absorption line and the line of sight (LOS) velocity due to the Zeeman and Doppler effects. Arriving to these quantities is possible by the inversion of the Radiative Transfer Equation (RTE). See del Toro Iniesta (2003) for more details on spectropolarimetry.

SO/PHI is the first spectropolarimeter on a deep space mission, facing an unprecedented dynamic environment and telemetry limitations. These challenges are met with a full and autonomous on-board data analysis system: it determines the instrument characteristics, applies them to the science data, then derives the targeted physical pa-

rameters. This system is implemented on a data processing unit with two Field Programmable Gate Arrays (FPGAs), reconfigured in flight to perform image processing functions (Fiethe et al. 2012; Lange et al. 2017), and a microprocessor running a data processing framework that combines these functions into pipelines (Albert et al. 2018). This contribution analyses errors induced by the on-board processing.

2. The on-board data analysis software

The science data processing comprises of preprocessing and RTE inversion. The preprocessing primarily corrects the images for the dark and flat field of the instrument, and does the polarimetric demodulation. Depending on science case and the instrument parameters determined at instrument commissioning, it may have additional steps (e.g. spatial cropping or deconvolution from image artifacts). The RTE inversion transforms the 24-image spectropolarimetric dataset into 5 images of interest: azimuth, inclination and magnitude of the magnetic field, the LOS velocity and the total intensity at continuum wavelength. See Cobos Carrascosa et al. (2016) for details on SO/PHI's RTE inversion scheme. To save FPGA resources, the preprocessing functions use fixed point number representation on 24.8 bits, while the RTE inversion is on 32 bits floating point.

The most basic preprocessing pipeline for a data set from an imaging spectropolarimeter contains dark and flat field correction and polarimetric demodulation:

$$S_{\lambda}(x, y) = D(x, y) \cdot [(I_{\lambda}^{obs}(x, y) - I^{dark}(x, y)) / I^{flat}(x, y)], \quad (1)$$

where "." denotes matrix multiplication, λ marks wavelength dependence, x and y are spatial dimensions. The Stokes parameters are contained in S , D is the demodulation matrix, I^{obs} contains the observed data in the four modulation states. The dark field of the sensor is I^{dark} , I^{flat} is the telescope flat field, neither depending on wavelengths and modulation states (may change for the flat field after instrument commissioning).

To implement Eq. 1, we use four blocks, combined into a pipeline (see Fig. 1). The raw data is integer, represented up to 22.8 bits after accumulation (14.8 assumed in test). As the exposure time is calibrated to fill a defined percentage of the detector full well, the recorded data is ideally represented. To process the data at the highest resolution, we shift the pixel values of these images to the top of the full range ($\times 2^9$), arriving to 23.8 bits (one bit is sign). This representation is the block interface, however some blocks re-scale the images to optimize the output accuracy.

We quantify the errors introduced through on-board processing by running the pipeline on a SO/PHI ground reference model and on a ground computer (using floating point in Python). The test data is from the Solar Dynamics Observatory / Helioseismic and Magnetic Imager (see Schou et al. 2012), run through the SO/PHI instrument simulator, SOPHISM (see Blanco Rodríguez et al. 2018). This data is modulated into measured intensities, then degraded with flat and dark field obtained during ground calibration (also used by the pipeline). We compare the results of the preprocessing after flat field correction, and polarimetric demodulation (no errors are expected from subtraction). The RTE inversion is done on a ground computer, with the He-Line Information Extractor inversion code (HeLIx⁺; see Lagg et al. 2004). HeLIx⁺ assumes a Milne-Eddington approximation of the atmosphere, the same as SO/PHI's on-board inversion scheme, however the profile fitting method is different. The differences in physical parameters only indicate the expected error induced by numerical inaccuracy.

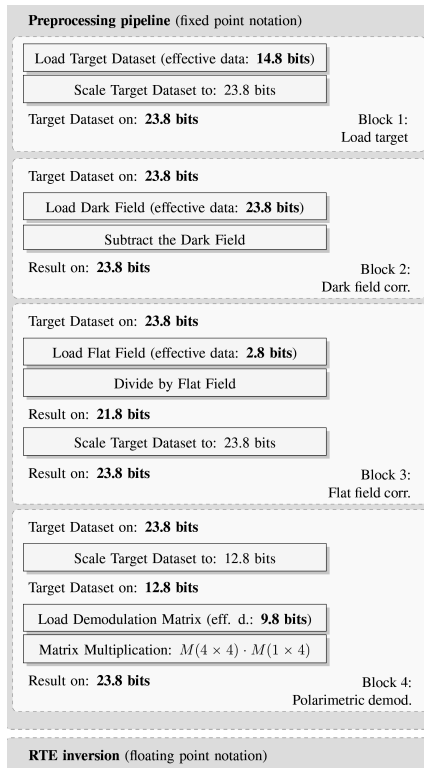


Figure 1. The studied pipeline. The preprocessing controls accuracy by scaling.

cies in preprocessing, due to the differences in the algorithms, the innate uncertainty in the results of the inversion.

3. Results

The errors from the division are below 10^{-3} (compared to the reference results), apart from a few outliers in the divisor, with Root Mean Square (RMS) around 5×10^{-5} .

SO/PHI requires the accuracy of the polarization signals (i.e. S) to be better than 10^{-3} . Figure 2 shows the error histogram at one wavelength sample. All pixels comply, with their RMS in the order of 10^{-6} , leaving a large margin for other error sources. The errors decrease from the previous step due to the nature of polarimetry: it calculates the difference between signals, partially canceling previous errors. Furthermore, the error in Q is larger than in the rest of the Stokes images, due to a small term in D .

The errors after RTE inversion are only an indication of what is expected due to numerical inaccuracies. The RMS error of the magnetic field strength, azimuth, inclination (calculated in a region with strong signals) and LOS velocity (calculated in the entire solar disk) due to numerical errors are 33.64 G, 1.92° , 2.56° , and 19.00 ms^{-1} , respectively. The inversion process for this type of data set, statistically, has error RMS

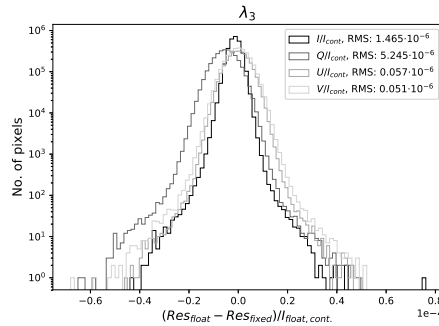


Figure 2. Histogram of polarimetric errors, showing requirement compliance.

21.9 G, 1.37° , 1.34° and 14.5 ms^{-1} , respectively. What is introduced on top of this by the numerical inaccuracies amount to 53%, 40%, 91% and 31% of the inversion error.

4. Conclusions

SO/PHI is the first instrument of its kind to perform on-board data analysis, including data preprocessing and the inversion of the RTE. These steps use computationally demanding image processing functions, implemented on FPGAs. The fixed point number representation in the on-board preprocessing was motivated by resource limitations.

The errors induced by the preprocessing conform with requirements, with a good margin for other sources. This is achieved by keeping full control over data accuracy, a significant overhead. Errors in Fourier domain processing are currently being analyzed.

Acknowledgments. Workframe: International Max Planck Research School (IMPRS) for Solar System Science. Solar Orbiter: ESA, NASA. Support grants: DLR 50 OT 1201, Spanish Research Agency ESP2016-77548-C5, European FEDER. Data: NASA/SDO HMI science team.

References

- Albert, K., Hirzberger, J., Busse, D., et al. 2018, in Proc. SPIE, vol. 707, 10707
- Blanco Rodríguez, J., et al. 2018, The Astrophysical Journal Supplement Series, 237, 35
- Cobos Carrascosa, J. P., et al. 2016, in Proc. SPIE, vol. 9913, 9913
- del Toro Iniesta, J. C. 2003, Introduction to spectropolarimetry (Cambridge university press)
- Fiethe, B., Bubenhausen, F., Lange, T., Michalik, H., Michel, H., Woch, J., & Hirzberger, J. 2012, in NASA/ESA Conference on Adaptive Hardware and Systems, 31
- Lagg, A., Woch, J., Krupp, N., & Solanki, S. K. 2004, Astronomy and Astrophysics, 414, 1109
- Lange, T., Fiethe, B., Michel, H., Michalik, H., Albert, K., & Hirzberger, J. 2017, in NASA/ESA Conference on Adaptive Hardware and Systems, 186
- Müller, D., Marsden, R. G., Cyr, O. S., et al. 2013, Solar Physics, 285, 25
- Schou, J., Scherrer, P. H., Bush, R. I., et al. 2012, Solar Physics, 275, 229
- Solanki, S., del Toro Iniesta, J., Woch, J., et al. 2018, Submitted to Astronomy and Astrophysics

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Optimization of Aperture Photometry in the Chandra Source Catalog

Christopher Allen, Joseph B. Miller, and Francis A. Primini

Smithsonian Astrophysical Observatory, Cambridge, Massachusetts, United States; ceallen@cfa.harvard.edu

Abstract. The Chandra Source Catalog has identified over 315,000 unique sources, across 10,382 observations. The aperture photometry system was created to characterize energy and photon fluxes for these observed sources by fitting a point-spread function matrix and observed counts to calculate actual intensities. To support overlapping point-spread functions of adjacent sources, multiple sources are fitted simultaneously. Sources are fitted per observation, across all observations to create master properties, and across subsets of similar observations, using a Bayesian blocking algorithm applied to the per-observation aperture photometry results.

Populating the catalog, aperture photometry is run a great many times - approximately 2.9 million times on the current detection list. As such, optimization in the dimensions of both time and memory is crucial. Herein we will discuss the challenges and decisions made as we moved from requirements analysis, prototyping, and development into production.

1. Introduction

The Chandra Source Catalog (Evans et al. 2010) is intended to be the definitive catalog of X-ray sources detected by the Chandra X-ray Observatory over its lifetime. The 2.0 version of the catalog release expands on the source properties characterized, and the aperture photometry systems play a key part in this. Aperture photometry produces four key properties: energy flux, photon flux, net counts, and count rate. These properties are computed for each observation a source is detected in, for each energy band and for both the detect apertures and the ecf90 apertures (an ecf90 aperture is defined to be 90% of the point source's emission as captured by that observation). The aperture photometry characterization consists of producing a marginalized probability distribution function (mPDF) along with identifying the mode and upper and lower 68% confidence intervals. The mPDFs are grouped into blocks of similar intensity, using a Bayesian blocking algorithm. Single intensities are then determined for each block, using data from all observations comprising the block, and allowing a master source calculation with a generally tighter confidence interval width. Both flux-ordered and time-ordered blocks are produced, all of which are characterized via aperture photometry.

Each of the individual fits is performed on either a single source or a group of adjacent sources. The process involves computing a joint posterior probability distribution (jpPDF), defined as the product of the Poisson likelihoods of observed source and background counts for a given source and background intensity. We assume positive, but otherwise uninformative priors, and in most cases the best-fit intensities are

$$\begin{bmatrix} Psf_{s_1,r_1} & Psf_{s_1,r_2} & \dots & \dots & Area_{s_1} \\ Psf_{s_2,r_1} & Psf_{s_2,r_2} & \dots & \dots & Area_{s_2} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & Psf_{s_n,r_n} & Area_{s_n} \\ Psf_{s_1,bkg} & Psf_{s_2,bkg} & \dots & Psf_{s_n,bkg} & Area_{bkg} \end{bmatrix} \begin{bmatrix} Inten_{s_1} \\ Inten_{s_2} \\ \dots \\ Inten_{s_n} \\ Inten_{bkg} \end{bmatrix} = \begin{bmatrix} Counts_{s_1} \\ Counts_{s_2} \\ \dots \\ Counts_{s_n} \\ Counts_{bkg} \end{bmatrix} \quad (1)$$

Figure 1. Formula to simultaneously solve N adjacent source intensities, given PSFs and observed counts for sources and backgrounds. Background noise and source counts that are detected in the background or a neighboring source are accounted for by computing each source's PSF fraction that falls in the background or neighboring regions.

essentially the Maximum Likelihood estimates (MLE). We calculate these MLEs using a matrix of point-spread fractions for each source and aperture, computing observed counts and exposure time from the observation, and solving the resultant equation to find the source intensities (Figure 1).

The per-observation, per-block, per-shape, per-band, per-mode multiplication effect means that in the current production run, the Aperture Photometry system will run approximately three million times, given the current size of the catalog. As such, optimization both for memory and time is critical to a timely production of the catalog.

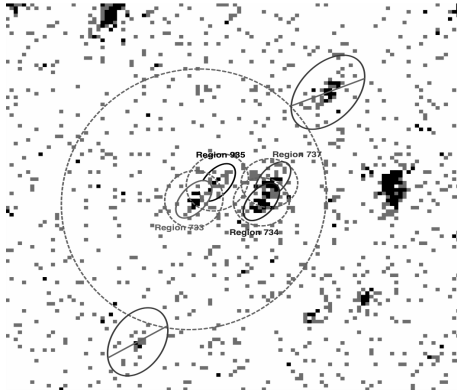


Figure 2. Several adjacent sources bundled together for fitting

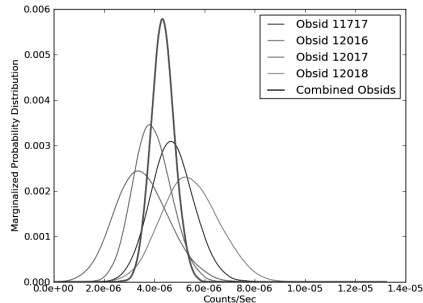


Figure 3. Fitting the same source across multiple observations narrows the confidence interval

2. Creating a Robust Data Model

The Aperture Photometry system offers challenges in that some aspects are very I/O intensive, and some are very processor intensive. A goal in the initial design was to separate the two, allowing our processing systems to be more flexible in allocating resources, and make it easier to perform follow-up analysis using small interim products rather than raw inputs.

The bundle is the fundamental level at which aperture photometry is performed—a group of sources close enough together in a single observation for their PSFs to appreciably impact each other (Figure 2). Calculating the point-spread-function (PSF) fractional matrix and observed counts for a single bundle requires region files, event files, single-source PSF maps, and exposure maps. If the raw inputs were used for each of the fits and PDF calculations, these input products would be needed multiple times. To prevent multiple redundant I/O operations, we added a pre-processing step that synthesizes the raw inputs into a single small file for any given observation and bundle. This file contains the fractional PSF matrix for all the sources and background, counts across the different modes, and initial estimates that are used as the starting location for the fitting algorithms.

3. Hypermesh vs Markov chain Monte Carlo

Initial implementations of the fitting algorithm called for the construction of an N-dimensional hypermesh to explore the jpPDF, where N was equal to the number of independent variables in the fitting (number of sources plus background). Each dimension's range was defined by the MLE estimate for the corresponding source or background intensity, bounded by its standard deviation. Over this range, 200 evenly spaced intensity samples defined. The hypermesh was then populated with the product of the Poisson likelihoods for obtaining observed source and background aperture counts, given the sampled source and background intensities. The mPDF for each source was then determined by numerical integration over all other dimensions of the hypermesh.

While suitable for small datasets, this implementation suffered for large N, due to the geometric memory and time growth required to store and populate the hypermesh. By replacing the hypermesh with a combination of sherpa's fitting package (Fruscione et al. 2006) to optimize initial parameters, and Markov chain Monte Carlo to sample a posterior probability distribution, the application can scale up to much larger datasets.

Moving from a mesh to a sampling-based distribution created a new hurdle, in that several downstream elements of the pipelines assumed a continuous mPDF as input. To recover a continuous distribution, we applied a kernel smoothing function (Wand 1995) to the distribution of MCMC draws for each parameter. Initially developed in native Python, we migrated to Cython to create a high-performance Python extension module with minimal effort. When the Cython module did not exhibit the expected performance improvements, we used the built-in profiling tools to identify poorly implemented interfaces between the C and Python layers. After addressing these implementation errors, we achieved a 50x speedup over native Python code - approximately the same performance achieved by a pure C module as shown in later testing.

4. Profiling

Using a variety of modules, preliminary versions of the tool were analyzed for run time and memory usage. Python's built in cProfile module allows checking of the whole program via command-line options, and individual functions via decorator tagging. The line_profiler package (https://github.com/rkern/line_profiler) allows fine-grain review of individual functions. Instrumentation of individual functions is critical, as algo-

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Utilizing Conda for Fermi Data Analysis Software Releases

Joseph Asercion

NASA Goddard Space Flight Center, Greenbelt, Maryland, USA;
joseph.a.asercion@nasa.gov

Abstract. The Fermi Gamma-Ray Space Telescope mission provides, via the Fermi Science Support Center (FSSC), a suite of data analysis tools to assist the high energy astrophysics community in working with Fermi data. For many years these tools were distributed via both precompiled binaries and source tarball downloads on the FSSC's website. Due to the complexity of the tools and restrictions on development the downloads carried with them a large complement of third-party software which often caused package conflicts on user's machines and bloated the size of the complete analysis package. To alleviate these problems the Fermi development team has decided to update the distribution pipeline to utilize the Conda package management system. This has allowed the development team to greatly reduce the software package size, eliminate a large category of bugs which once were prevalent, and target a decrease in software update turnaround/release time. In this poster, I will outline the process the development team took to convert our legacy codebase into a Conda compatible form.

1. Introduction

In order to facilitate Fermi data analysis by the high energy astrophysics community the Fermi mission, through the FSSC, provide a suite of data analysis tools which are designed to assist with common forms of gamma ray astronomy data analysis. These tools, known as the Fermitools in our most recent release, have been in development since before the observatory launched in 2008 and have been added to and updated since. The large size and complexity of the tools as a whole have necessitated a complicated release process in the past.

Previous to the October 2018 release of the Fermitools, the suite was made available to the public via downloads of precompiled binaries or source code tarballs. fer (2018b) Due to design constraints at the start of the mission, the downloads contained both the code for the tools themselves and a large number of third party packages (such as ROOT, Python, FFTW, and Xerxes) which the tools depended on. The idea was to make the software as self-contained as possible to increase portability, however, this bloated the software package greatly and made the suite prone to library mismatch and package collision errors with software already present on the user's machine.

To help reduce both the size of the software packages and the number of collision errors, it was decided that the tools would be distributed via a package management system. In addition to reducing errors on user's machines and increasing portability, offloading dependency management to a third party the size of the code base Fermi developers would need to manage would decrease greatly allowing more developer resources to be devoted towards maintaining and improving the analysis code itself.

After exploring several options, it was decided that the Conda Package Manager would be used to manage and distribute the Fermitools to the public.

2. Why Conda?

Previous to the decision to move to a package management system there were ongoing discussions about replacing the version of Python 2.7 that was distributed with the tools with Anaconda Python due to its prevalence in the Fermi analysis community and the number of modules built into Anaconda which were already used by the tools. This proposed change along with the wide use of Python for Fermi data analysis in general made the Conda Package Manager a natural choice.

Conda also allows for easy packaging and distribution of the analysis tools via the Conda Build utility and the Anaconda Cloud, respectively. The ability to take advantage of the Conda-Forge GitHub organization's CI toolchain for long term maintenance of FSSC maintained dependencies was viewed as another major benefit.

2.1. What is Conda?

Conda is the native package management system in Anaconda and Miniconda Python distributions. Developed and maintained by Anaconda, Inc (formerly Continuum Analytics), it is released under the Berkeley Software distribution license. Conda is open source, cross-platform, and language-agnostic. con (2017)

In addition to package management functionality Conda also has environment management capability. Conda environments allow users to create self-contained installation environments for software. This prevents software installed by Conda from conflicting with software that is already installed on the user's machine or in separate Conda environments. con (2017) A major benefit of this functionality is that software with conflicting dependencies can coexist within the same Anaconda installation so long as they are located within different Conda environments. Unlike pip, Conda can also install different versions of Python. This is incredibly useful in situations where a user needs to have access to software which requires either Python 2 or Python 3 without having to install/configure multiple versions of Python in their user environment.

Conda pulls precompiled software from channels hosted on the Anaconda Cloud. While the Anaconda channel is default, users can specify what channels they would like Conda to search and the channel priority in the search order. Due to the complex set of dependencies that the Fermitools uses this is a critical feature. The FSSC supplies a required channel list to users who wish to install the Fermitools in order to ensure that the correct dependencies of the software are selected. fer (2018b)

3. Conda Packaging

Conda uses a built-in utility name Conda Build to package compiled binaries in a format suitable for upload to the Anaconda Cloud. To provide Conda Build with the instructions it needs to properly assemble the target software two files (at a minimum) need to be provided to the utility as part of a 'recipe': meta.yaml, which defines the build and runtime meta data of the package being built and build.sh, a bash script which directly instructs Conda build in how to compile and assemble the target software.

To assemble the binaries Conda Build uses a temporary directory tree that it creates within Anaconda's directory structure to hold the source code and various build products generated during the execution of the build.sh script. Once compilation and testing (as outlined by the meta.yaml file) are complete Conda Build compresses the binary build products along with several files containing package metadata and produces a tarball. This tarball can then be uploaded to a specified Anaconda Cloud channel for distribution to the community.

4. Transition Process

After consolidating the Fermitools codebase a number of changes needed to be made to the tools to accommodate the Conda Build system. Despite being compiled and packaged as a monolithic software distribution the tools themselves are maintained as separate subpackages which depend on one another. To address this issue, the build.sh script included in the Fermitools recipe was modified to call a customized tool which had been designed specifically to properly conduct checkouts of Fermi-lat git organization software. After checkout, Conda Build is instructed to compile the Fermitools using a specialized Scons command. Special handling for additional data files is also included within the build.sh to ensure that necessary models and reference code is included in the packaged tarball.

The meta.yaml file lists the dependencies that are required installations for the Fermitools to run properly. All of the dependencies listed for both the build and run stages are pinned to specific versions in order to prevent possible issues caused by the third party package maintainers updating their code. The vast majority of the Fermitools dependencies are retrieved for the Conda-Forge Anaconda channel. Conda-Forge is a GitHub organization which contains a number of Conda recipes that are automatically tested and built using a Continuous Integration chain (AppVeyor, TravisCI, and CircleCI) to build, test, and make available for download a number of different community maintained packages. Scopatz (2018) The FSSC maintains several of the Fermitools dependencies within Conda-Forge to take advantage of the CI tools which the organization offers.

Necessary data files, and a few irregular software dependencies, are distributed by the FSSC on a separate 'fermi' Anaconda Cloud channel. fer (2018a) The decision to separate out the data files from the primary software package allows for much more flexibility and ease in updating the data and streamlines the organization of the Fermitools source code.

5. Result

The Fermitools is currently available to public download from the Fermi anaconda channel (for instructions on how to install the Fermitools see the section 'Obtaining the Fermitools'). As noted above, stripping third party dependencies from the Fermi

	ScienceTools v11r5p3	Fermitools (Tools+Data)
Mac Binary	1.44 GB (Darwin 16.7)	78.4MB+497.8MB
Linux Binary	1.94 GB (Scientific Linux 7)	156.5MB+497.8MB

data analysis suite substantially reduced the size of the software package that is directly

managed and distributed by the FSSC. While the dependencies must still be managed directly and version compatibility must still be monitored/tested the use of the Conda package management system simplifies this process and allows developers to focus more on the core tools.

6. **Obtaining the Fermitools**

To install the Fermitools first a version of Anaconda Python needs to be installed. Either Anaconda or Miniconda will do, however, the FSSC recommends Miniconda as it is more lightweight. It is **highly recommended** that the Python 2.7 version of Anaconda/Miniconda be installed. The tools **require** Python 2 at this time and while it is possible to install a different version of Python in the tool’s Conda environment for most users this is an unnecessary complication.

At the end of installation Conda will ask to append the its location to the front of the user’s PATH in their bashrc file. Users who prefer tcsh/csh must perform an additional step to ensure that their Conda installation works properly in their shell:

```
source </path/to/conda>/etc/profile.d/conda.csh
```

must be appended to their `/.cshrc` or `/.tcshrc` file. `</path/to/Conda>` is replaced with the path to the top level directory of the Anaconda installation (typically named something like "Miniconda2"). This is because, while Anaconda natively supports the BASH shell, it does not yet automatically complete setup for csh/tcsh.

After completing installation of the tools, a user need only run the command

```
Conda create -name fermi -c Conda-forge -c fermi fermitools
```

To create a Conda environment named ‘fermi’ and install the latest release version of the Fermitools into it (currently v1.0.0). Once this process is complete, the user needs only to activate the environment with the appropriate command for their shell:

Bash	Csh/Tcsh
source activate fermi	Conda activate fermi

After activation the user will have access to the full suite of Fermitools installed in the ‘fermi’ environment. fer (2018b)

References

2017, Conda User Guide, Anaconda, Inc. URL <https://conda.io/docs/user-guide/index.html>
 2018a, Fermi Organization Anaconda Channel, Fermi-LAT Collaboration. URL <https://anaconda.org/fermi>
 2018b, Fermitools Wiki, Fermi Science Support Center. URL <https://github.com/fermi-lat/Fermitools-conda/wiki>
 Scopatz, A. 2018, Conda-Forge Documentation. URL <https://conda-forge.org/docs/>

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Streamlining Pipeline Workflows: Using Python with an Object-Oriented Approach to Consolidate Aggregate Pipeline Processes

M. K. Brown,¹ J. A. Mader,¹ G. B. Berriman,² C. R. Gelino,² M. Kong,²
A. C. Laity,² J. Riley,¹ L. Rizzi,¹ and M. A. Swain²

¹*W. M. Keck Observatory, Kamuela, HI, USA; mbrown@keck.hawaii.edu*

²*CalTech/IPAC-NExScI, Pasadena, CA, USA*

Abstract. The Keck Observatory Archive (KOA), a collaboration between the NASA Exoplanet Science Institute and the W. M. Keck Observatory, serves science and calibration data for all current and retired instruments from the twin Keck Telescopes. In addition to the raw data, we publicly serve quick-look, reduced data products for four instruments (HIRES, LWS, NIRC2, NIRSPEC and OSIRIS), so that KOA users can easily assess the quality and scientific content of the data. In this paper we present the modernization of the Data Evaluation and Processing (DEP) Pipeline, our quality assurance tool to ensure science data is ready for archiving. Since there was no common infrastructure for data headers, the DEP pipeline had to evolve to accommodate new instruments through additional control paths each time an instrument was added or upgraded. Over time, new modules to assist with the processing were added in a variety of languages including IDL, C, CSH, PHP, and Python. The calls to multiple interpreters caused a lot of overhead. This project was an initiative to consolidate the DEP pipeline into a common language, Python, using an object-oriented approach. The object-oriented approach allows us to abstract out the differences and use common variables in place of instrument-specific values. As a result, new instruments only need a modified subclass with the differing values in order to work with the pipeline. By consolidating everything to Python, we have seen an increase in efficiency, ease of operation, and ease of maintenance.

1. Introduction

As time moves forward, technology and software change. New languages are introduced, new standards of coding are codified, new packages are released for existing languages, and new methods of data transfer and analysis are devised—just to name a few. One of the more prolific changes to programming and software development was the introduction of object-oriented programming which encapsulates all the relevant information about something into a single entity. This change allowed programmers to keep better track of information they were interested in leading to more efficient and maintainable software! Classes and class inheritance allow similar objects to be created without having to reinvent the entire wheel each time which means developers save time when adapting a system for new functionality.

In order to take advantage of these new abilities, new programming languages are developed. Librarians Pigott & Axtens (1995) have estimated that since the creation of

the first primitive programming languages in the early 1950s, 8,945 different languages have been created and published. While there are many languages designed for a specific purpose, such as R and S for statistical analysis or COBOL for business-oriented ventures, one in particular has taken a major foothold in the science and astronomy community: Python. The ease with which it can be learned and the wide availability of Python on the different computing platforms (Windows, *nix, and iOS) made Python a solid choice for experiment and analysis reproducibility regardless of one's working environment. As the science community became more involved with Python, packages such as NumPy for numerical analysis, SciPy for scientific analysis, and AstroPy for astronomical data analysis were developed to form a core set of scientific packages available to the science community at-large.

The Keck Observatory Archive (Berriman et al. 2014) was initially developed to archive data from the HIRES spectrograph on the Keck I telescope, with the hopes of eventually expanding to include data, past and future, from the other Keck instruments. This expansion has come to fruition, but quite a bit of "technical debt" has accrued within the data processing software. As a result, the software is a combination of several languages and maintenance and reprocessing of data is time consuming. An effort is underway to consolidate the processing software into a single, modern language, with goals to have it easily maintained, easily add instruments and easily reprocess data. This paper presents this new design and implementation.

2. Background of the Keck Observatory Archive

The Keck Observatory Archive requires files to be processed so that the metadata has all the keywords required for the database ingestion process which we call the Data Evaluation and Processing (DEP) pipeline. When it was first started in 2002, NASA only required HIRES data to be archived in KOA. With the initial success of KOA, NASA saw fit to expand the Archive to the rest of WMKO's ten instruments. IDL was the language of choice to edit the headers of the FITS files. However, as more instruments were added, the software base became cluttered with if statements to handle each instrument case. Furthermore, multiple instruments have more than one type of image they can capture which needed to be known for proper ingestion of the data. Some of the subsystems were rewritten in new languages, like PHP, to better handle things like string or array manipulation. These problems led to a growing concern that DEP would become unmanageable and unmaintainable as more instruments were added to the telescopes. This concern has been met with an internal initiative to consolidate the DEP Pipeline into a single language, the natural choice being Python.

3. The Consolidated DEP for KOA

Moving everything to Python was an obvious choice. It was clear that the astronomy community-at-large had decided to adopt it for their needs. Python offers a simple-to-use syntax without the complexity of strong typing and is based on the C programming language. It also has strong plotting capabilities and statistical analysis packages. This versatility will be useful for other projects beyond the scope of DEP.

Choosing to go with an object oriented approach was a decision that was made with regards to future implementations. Currently, the DEP Pipeline is run the next

morning after an observing run. Changes are being made to ingest the data in real time which is currently not possible with the procedural version of DEP. The procedural version assumes it will find every science file necessary for the processing and the work required to make it run file by file is actually less than the work to redesign the system with an object-oriented approach in mind. In this approach, each file is treated as an object and checked to verify that it has the right metadata. There are some consistencies between instruments, so we have a general outline that all instruments inherit from. Then, to deal with the differences, we create subclasses that can handle the differing keyword values. The DEP pipeline only needs to use the generalized variables from the object and doesn't worry about the underlying values. This approach makes it very easy to add new instruments as we just replace the values while keeping the variables the same. Rather than going to each script and adding more if statements to handle it, a new subclass file can be created and accessed through the pipeline with an import statement. This saves a lot of time for development and debugging because all the changes are now centralized.

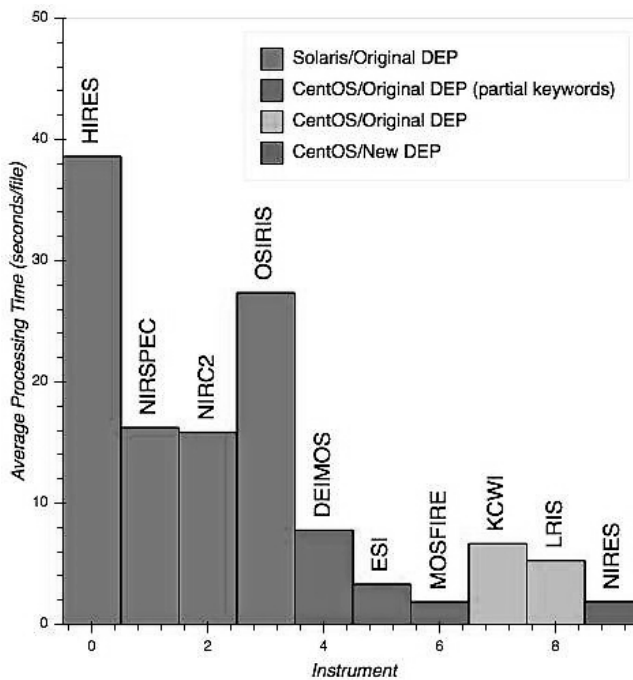


Figure 1. Processing times of DEP with different implementations.

4. The Results

With the implementation of the Python pipeline for the NIRES and MOSFIRE instruments there have been improvements in the processing time. NIRES was the first instrument implemented with the new Python design and currently has the lowest processing

time of any of the instruments. The second instrument implemented, MOSFIRE, has shown the processing time to stay about the same as its IDL counterpart. However, more processing is being done in the new design as the IDL version only verified a few basic keyword parameters. A full keyword check is now being used in the Python version. The increase in work with the stability of the processing time shows that the new Python pipeline is more efficient.

The new design has also simplified maintenance and the steps required to reprocess data. The IDL version required each individual step to be run and the output from those steps were used by subsequent ones. The new design uses the same command to run all or part of the processing steps, with optional start and stop states supplied on the command line. This greatly increases efficiency when testing and reprocessing data.

Another added benefit is that the new python pipeline does not rely on IDL anymore. All FITS handling is done through the `astropy.io.fits` package. This saves operating costs as IDL licenses are no longer needed for the DEP processing. It also saves resource overhead as the pipeline no longer has to start up IDL workspaces every time an IDL procedure is called. There is an incurred risk as `astropy` is an open source software package which could be updated at any time. However, this risk could easily be mitigated by freezing the production version of `astropy` at a verified, validated, and trusted version which is backed up locally.

5. Conclusions and Moving Forward

Software is evolving at an incredible rate. Every year, new languages are released and some are even useful. It is easy to get stuck in the ebb and flow of nightly operations and the technical debt that is accrued can become unmanageable. We must make the time and put forth the effort to keep our operations current so that we can capitalize on these advancements in technology and produce the most efficient results. This paper is an example of how converting old pipelines with pieces in multiple languages into a single language to remove technical debt can increase the efficiency, reduce the cost, and ease the maintainability of existing software systems. This new object-oriented approach with Python will provide a basis for future instrument additions to KOA allowing for rapid development and implementation of the new sub-classes. It is also the basis for future improvements such as real-time ingestion and full-keyword headers for science files which will utilize the developed sub-classes to act as a control for the differences between the instruments.

Acknowledgments. KOA is a collaboration between the W. M. Keck Observatory (WMKO) and the NASA Exoplanet Science Institute (NExSci). Funding for KOA is provided by NASA under award No. 80NSSC18M0066. WMKO is operated as a scientific partnership among the California Institute of Technology, the University of California and NASA.

References

- Berriman, G. B., Gelino, C. R., Goodrich, R. W., Holt, J., Kong, M., Laity, A. C., Mader, J. A., Swain, M., & Tran, H. D. 2014, in *Software and Cyberinfrastructure for Astronomy III*, vol. 9152, 91520A. 1408.0835
- Pigott, D. J., & Axtens, B. M. 1995, *History of programming languages*. URL <http://hop1.info>

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

GMRT Archive Processing Project

Shubhankar Deshpande¹, Yogesh Wadadekar², Huib Intema³, B. Ratnakumar²,
Lijo George², Rathin Desai², Archit Sakhadeo², Shadab Shaikh²,
C. H. Ishwara-Chandra² and Divya Oberoi²

¹*Carnegie Mellon University, USA; shubhand@cs.cmu.edu*

²*National Centre for Radio Astrophysics, TIFR, Post Bag 3, Ganeshkhind,
Pune 411007, India*

³*Leiden Observatory, Leiden University, Niels Bohrweg 2, 2333 CA, Leiden,
The Netherlands*

Abstract. The GMRT Online Archive now houses over 120 terabytes of interferometric observations obtained with the GMRT since the observatory began operating as a facility in 2002. The utility of this vast data archive, likely the largest of any Indian telescope, can be significantly enhanced if first look (and where possible, science ready) processed images can be made available to the user community. We have initiated a project to pipeline process GMRT images in the 150, 240, 325 and 610 MHz bands. The thousands of processed continuum images that we will produce will prove useful in studies of distant galaxy clusters, radio AGN, as well as nearby galaxies and star-forming regions. Besides the scientific returns, a uniform data processing pipeline run on a large volume of data can be used in other interesting ways. For example, we will be able to measure various performance characteristics of the GMRT telescope and their dependence on waveband, time of day, RFI environment, backend, galactic latitude etc. in a systematic way. A variety of data products such as calibrated UVFITS data, sky images and AIPS processing logs will be delivered to users via a web-based interface. Data products will be compatible with standard Virtual Observatory protocols.

1. Introduction

The Giant Meterwave Radio Telescope (GMRT, Swarup et al. 1991) is a low frequency radio interferometer operating at a site 80 km north of Pune, India. Since 2002, it has been operated as an international open access facility by India's National Centre for Radio Astrophysics. All interferometric observations carried out with the GMRT have been carefully archived over the years using several different tape and disk based storage technologies (see Fig.1). Raw interferometric visibilities were made available to the international user community, on request, via DVDs until 2009. Thereafter, all raw data were made accessible for search and download via a password authenticated, web-based interface, the NCRA Archive and Proposal handling System (NAPS)¹

¹<http://naps.ncra.tifr.res.in/goa>

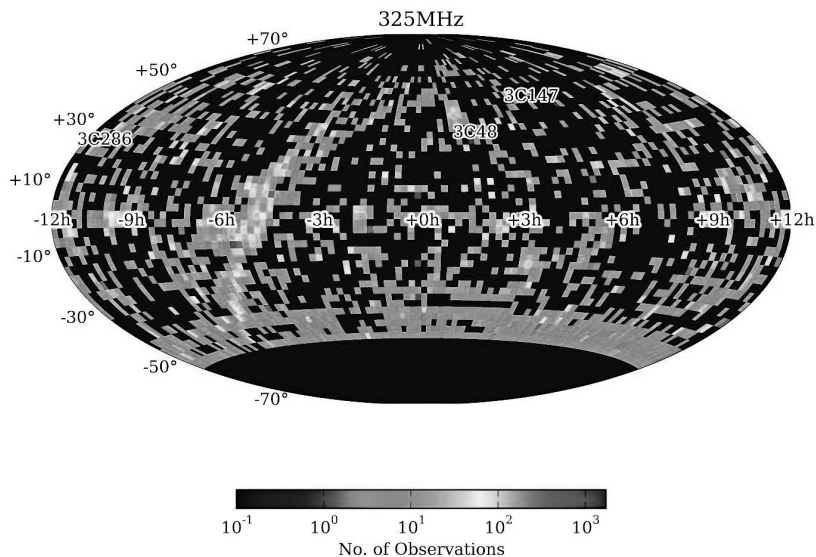


Figure 1. Observations made with the GMRT in equatorial coordinates at 325 MHz, with the sky pixellated into a 3x3 degrees grid for each datapoint. Much of the sky north of the declination limit of the GMRT has been covered. The 3 primary flux calibrators - 3C 48, 3C 147 and 3C 286 have been observed on numerous occasions. Some part of the galactic plane has also been extensively observed. These raw data will be processed into images by our project.

The NAPS system now hosts over 120 TB of data and delivers them to the GMRT user community located in about 40 countries worldwide. In 2018, we saw an average of about 55 data requests per month with an average size of 50 GB per request. Despite this high level of usage of the archive, the scientific utility of these data is greatly limited by the large effort required to transform these raw visibilities into science ready images. To address this situation, we have initiated an effort to generate pipeline processed continuum images for GMRT data. We are looking to provide users with “first look” (worst case) and “science ready” (best case) images for as many GMRT observations as possible. A “certifiably bad” tag on data is also useful, because it helps convince the time allocation committee that fresh observations are warranted.

2. Imaging the GMRT Archive

Presently, there is no standard or official pipeline for processing GMRT data. However, there are several scripts and pipelines developed by different users, some of which are publicly available. One of the most sophisticated, publicly available, pipelines for processing data from the GMRT is the Source Peeling and Modelling (SPAM) pipeline developed by H. Intema (Intema et al. 2017). It was used to successfully pipeline process

about 2000 hours of GMRT data from the TIFR GMRT Sky Survey (TGSS ADR², Intema et al. 2017). SPAM is a Python module that provides an interface to AIPS via ParselTongue (Kettenis et al. 2006) and ObiTalk (Cotton 2008). ParselTongue provides access to AIPS tasks, data files (images & visibilities) and tables. SPAM extensively uses several other popular Python modules like numpy and scipy. Data reductions are carried out by well-tested Python scripts that executes AIPS tasks directly or via high-level functions that make multiple AIPS or ParselTongue calls. SPAM now also includes a fully automated pipeline for reducing legacy GMRT observations at 150, 235, 325 and 610 MHz.

3. Building Our Compute Infrastructure

We used a simple Beowolf cluster architecture for our processing. A master node acts as a fileserver for the compute stack - AIPS, SPAM, ObiT and Parseltongue which is NFS exported to a set of compute nodes (simple headless desktop computers) which have a standard Ubuntu 16.04 server installation plus some additional software libraries installed via Tentakel from the master node. Ganglia was chosen as the tool for monitoring cluster status. After some experimentation, we found that using a Docker container to install our software stack on each compute node was more efficient. We began processing with a four node cluster which was gradually expanded to include about 30 headless desktops. Even with this modest hardware, it is possible to process about 5 months of GMRT data in about a month. The raw data archive is hosted on a Dell EMC Isilon system from where it is NFS exported to our data processing cluster. After processing, the outputs are copied back onto the Isilon system for long term storage and disaster recovery compliant backup.

4. Data Processing and Delivery

The SPAM pipeline is designed to be used interactively by a single user and each processing instance runs as a single thread on the CPU. We wrote a set of Python and bash scripts to make SPAM operate in non-interactive fashion and to run multiple processing threads simultaneously on each multicore computer in our cluster.

We realised quickly that keeping track of the processing was very cumbersome since data processing rarely progressed linearly. Failures could happen due to poor data quality or due to some limitation in SPAM. It was important to bookkeep all of these so that we could get an accurate picture of the current status of the processing for each observation and to gather statistics on failure situations. We have developed a comprehensive database schema to keep track of the processing. Scattered throughout the SPAM processing are read and write calls to the database recording successes and failure and metadata on them. This database is also critical in determining which outputs are ready to be delivered to users after some automated and manual quality control. The database will also prove useful in analysis of the long term trends at the observatory in terms of the evolving telescope characteristics with manmade radio frequency interference and the ionospheric environment.

²<http://tgssadr.strw.leidenuniv.nl>

For datasets where the processing is successful (see Fig.2) a variety of data products such as calibrated UVFITS data, sky images, AIPS processing logs are generated and are now being integrated into the NAPS system for delivery. These will become visible to NAPS users as additional value added data products which will be compatible with standard Virtual Observatory protocols.

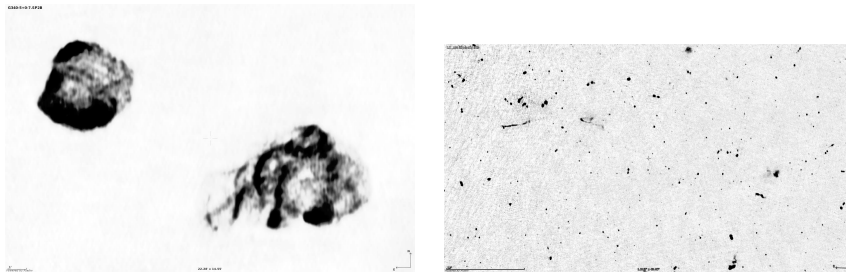


Figure 2. Left: Two supernova remnants near the galaxy centre observed at 325 MHz with the GMRT. Right: Portion of a 325 MHz image of the Lockman Hole field. The extended filamentary sources at top left are cluster radio relics. > 5000 radio sources are seen over $\sim 12 \text{ deg}^2$. These are just two of the thousands of radio images being produced by the GMRT Archive Processing Project.

5. Future Plans

The current SPAM pipeline only works on legacy GMRT data. For data which is now streaming from the upgraded GMRT (uGMRT, Gupta et al. 2017), with seamless frequency coverage and large bandwidth, a different pipeline would be needed. It would also be tremendously useful if the data processing can be done in near real time, so that authorised users can view the processed images from their own observations, within a few days. We are currently exploring the software and hardware enhancements that are necessary to enable these exciting possibilities over the next few years.

References

- Cotton, W. D. 2008, *PASP*, 120, 439
- Gupta, Y., Ajithkumar, B., Kale, H., Nayak, S., Sabhapathy, S., Sureshkumar, S., Swami, R., Chengalur, J., Ghosh, S., Ishwara-Chandra, C., Joshi, B., Kanekar, N., Lal, D., & Roy, S. 2017, *Current Science*, 113, 707
- Intema, H. T., Jagannathan, P., Mooley, K. P., & Frail, D. A. 2017, *A&A*, 598, A78. 1603. 04368
- Kettenis, M., van Langevelde, H. J., Reynolds, C., & Cotton, B. 2006, in *Astronomical Data Analysis Software and Systems XV*, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of *Astronomical Society of the Pacific Conference Series*, 497
- Swarup, G., Ananthakrishnan, S., Kapahi, V. K., Rao, A. P., Subrahmanya, C. R., & Kulkarni, V. K. 1991, *Current Science*, Vol. 60, NO.2/JAN25, P. 95, 1991, 60, 95

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Computational Astrophysics with Go

Pramod Gupta

Department of Astronomy, University of Washington, Seattle, Washington,
USA; psgupta@uw.edu

Abstract. Go is a relatively new open-source language from Google. It is a compiled language and so it is quite fast compared to interpreted languages. Moreover, it is based on a design principle of simplicity. In this paper, I discuss the suitability of Go for Computational Astrophysics based on using Go for Monte Carlo Radiative Transfer. I find that even though the language was not designed for scientific computing, its speed and simplicity make Go an excellent language for Computational Astrophysics.

1. Introduction

Computational astrophysicists have traditionally used compiled languages like C, C++ or Fortran for computation intensive research areas such as radiative transfer or N-body simulations. Since these languages are compiled, they have excellent run time performance. However, compared to interpreted languages like Python, they are relatively difficult to use due to lack of automatic memory management and run time checks.

The Go programming language is compiled so it has excellent performance. Moreover, it also has automatic memory management (garbage collection) and run time checks. Hence, one is lead to the question: Is Go a practical language for Computational Astrophysics? In this paper I propose an answer to this question based on my experience with using Go for Monte Carlo Radiative Transfer.

2. Go

Go was first released in 2009 and version 1.0 was released in 2012. Hence it is a relatively new language. The language is from Google but it is open source. Its creators are Robert Griesemer, Rob Pike, and Ken Thompson. (Ken Thompson is a Turing award winner and the creator of UNIX.). The standard book on Go is *The Go Programming Language* (Donovan & Kernighan (2015)). One of the authors of this book is the same Kernighan who wrote the *The C Programming Language* (Kernighan & Ritchie (1988)). Hence, Go has some very distinguished computer scientists associated with it.

Go is a compiled, statically typed language with built-in concurrency. Simplicity was an important design goal for the creators of Go. Due to the scale of Google, they were also concerned with improving performance by reducing compilation times and reducing running times. Also to increase reliability, Go has features which are common in interpreted languages like Python but which do not exist in compiled languages like C, C++ and Fortran. These are features such as automatic memory management (garbage collection) and run time checks (e.g., to detect array index out of bounds).

As noted in the previous paragraph, Go has several positive features. However, the question remains: Is Go a practical language for computational astrophysics? The only way to find out is to implement computational astrophysics code in Go. Hence, I implemented a Monte Carlo Radiative Transfer program in Go.

3. Monte Carlo Radiative Transfer

We consider Monte Carlo Radiative Transfer (MCRT) with scattering and absorption in spherical layers in an exoplanet atmosphere. An introduction to MCRT with polarization in a spherical geometry is given in Code & Whitney (1995). A parallel beam of photons is incident on the planet's atmosphere. An incoming photon enters the atmosphere with a Stokes vector $(I, Q, U, V) = (1, 0, 0, 0)$. It then travels an optical depth τ till it gets scattered or absorbed. The optical depth τ is given by $\tau = -\log(1 - \xi)$, where ξ is a random number between 0 and 1. The probability of the photon getting scattered is equal to the single scattering albedo. If the photon gets scattered, then the new Stokes vector is the product of a 4×4 matrix and the old Stokes vector. The random direction of scattering is dependent on a probability distribution based on the same matrix. A photon which reaches the surface of the planet gets absorbed or reflected back by the Lambertian surface. Hence, a photon either gets absorbed within the atmosphere, or it gets absorbed on the surface of the planet or it exits at the top of the atmosphere. If the photon exits at the top of the atmosphere then its exiting direction (θ, ϕ) is recorded. (Here θ and ϕ are the usual spherical polar coordinates.) For accurate results, a large number of incoming photons have to be simulated. Hence the program has loops with a large number of iterations.

As seen in the previous paragraph, the MCRT code uses multi-dimensional arrays, random numbers and large number of iterations for multiple loops. These are typical parts of a computational astrophysics programs. Hence, even though the present paper's conclusions are based on this program, they are more generally applicable to other computational astrophysics programs such as N-body programs.

4. Experience with Go

As noted above, simplicity was a design goal for the Go. Unlike most common languages of the last two decades, Go has no inheritance, no templates/generics, and no exceptions. The language was designed to be easy to learn for users with prior experience in the commonly used languages such as C, C++, Java, Python etc. Hence it is easy for computational astrophysicists to learn the language.

Go does automatic memory management (garbage collection) and run time checks such as array bounds checking. This increases the running time (compared to C, C++, and Fortran) but it also increases the reliability of the code. Since Go is a compiled language, the run time performance is very good compared to interpreted languages. Go is very strict about types. Even for numerical types, there is no promotion of `int64` to `float64`. For example if `x` is a `float64` variable and `y` is a `int64` variable then `x=x+y` will not compile. One must use `x=x+float64(y)`. This strict typing prevents various errors. Since each line is a statement, one does not get the kind of odd error messages which one can get in C and C++ by missing a semicolon. Since all variables are initialized to a default value of the type (e.g., default value for `int64` is `0`), one does

not get the run time errors due to uninitialized variables. Another feature of Go is that a function can return multiple values. This makes it possible to have a clear separation between input parameters and output parameters. For example, in the code below, `x` and `y` are the input parameters and `a` and `b` are the output parameters:

```
func some_function( x float64, y float64) (a float64, b float64){
//do some calculations
return a, b
}
```

Another function can call `some_function()`:

```
a, b = some_function( x, y )
```

Multi-dimensional arrays are essential to much of computational astrophysics. Passing multidimensional arrays to functions or subroutines is a common step in most programs. For passing such arrays to functions, C++ requires the sizes for all dimensions (except the first) to be known at compile time. Similarly, Go requires the sizes of all the dimensions to be known at compile time. However, Go has slices which are like dynamic arrays. Hence, one can use a slice of slices (i.e., dynamic array of dynamic arrays) to create a 2-dimensional array of `float64` (similar to `double` type in C and C++ and `double precision` type in Fortran). The code looks like below:

```
func make2DslicingFloat64( XSize int, YSize int) ([][] float64){
var i int
//make 2D slice (like array of arrays)
// Allocate the top-level slice.
a := make([][]float64, XSize)
// Allocate the next-level slices.
for i=0; i <XSize; i++ {
a[i] = make([]float64, YSize)
}
return a
}
```

Here `make2DslicingFloat64()` creates and returns a slice of slices. In the calling program, one would call `make2DslicingFloat64()` as below to make 3 x 3 matrix for the `float64` type:

```
var matrix1 [][]float64
matrix1 = make2DslicingFloat64(3, 3)
```

Since Go has garbage collection, the user does not need to remember to free the memory used by `matrix1`. The above code is similar to the Python numpy statement `a = numpy.zeros((3,3))`. Note that since Go does not have templates/generics, one would need to write another function `make2DslicingInt64()` for the `int64` type. We can pass `matrix1` to a function `trace()`:

```
trace(matrix1, 3, 3)
```


Also if we pass `matrix1` to a function then the function can modify the elements of `matrix1`. Built-in variables of type `string`, `int64`, `float64`, etc., and `struct` variables are passed by value so (just like in C) we must use pointers if we want to modify the variable in the function.

The Go compiler has fast compile times and it gives helpful and readable error messages. The executable produced by the Go compiler is statically linked which means that you can compile Go code on your machine, transfer the executable to another machine and it will run fine as long as both machines have the same operating system. The other machine does not need to have Go installed and it does not need any libraries other than those which are part of the operating system. This is very convenient if you develop your code on your desktop or laptop and then run the executable on a remote machine (e.g., a supercomputer). Go also has built-in features to run code concurrently.

Go has a standard code formatting tool `gofmt`. After formatting code with `gofmt`, everyone's code looks the same. This makes it easier to read code written by others. The language has excellent documentation and a large number of built-in libraries. The built-in packages for complex numbers and random numbers are especially useful for scientific computing. The external Gonum project has additional scientific computing packages. However, compared to Python, the number and scope of scientific computing packages is quite limited.

5. Conclusions

Go can be learned quickly by computational astrophysicists since they already know C, C++, or Fortran. Automatic memory management, run time checks, and strict typing reduce the code development time. Since it is a compiled language, the run time performance is very good. Due to Go's newness and since it is not targeted at scientific computing, there are a limited number of scientific computing libraries. Hence, Go may not be a feasible choice for projects which depend on specialized libraries. However, for projects which are not dependent on such libraries (such as Monte Carlo Radiative Transfer and N-body simulations), Go is an excellent language for computational astrophysics.

References

- Code, A. D., & Whitney, B. A. 1995, *Astrophysical Journal*, 441, 400
Donovan, A. A., & Kernighan, B. W. 2015, *The Go Programming Language* (Academic Press)
Kernighan, B. W., & Ritchie, D. M. 1988, *The C Programming Language* (Prentice Hall), 2nd ed.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

JWST Data Management Subsystem Operations: Rehearsing to Receive, Process, and Archive JWST Data

Catherine Kaleida, Anastasia Alexov, Mark Kyprianou, Faith Abney, and Matthew Burger

The Space Telescope Science Institute, Institution City, State/Province, Country; ckaleida@stsci.edu

Abstract. The James Webb Space Telescope (JWST) is a cornerstone in NASA's strategic plan, serving as the premier tool for studying the earliest stars and galaxies and for understanding the origins and future of the universe and the galaxies and solar systems within it. The Data Management Subsystem (DMS) is an integral part of the systems JWST needs to achieve these goals, as it serves as the interface between JWST and the astronomers who use it. We outline the JWST DMS Operations and detail the systems and tools that will be used to ensure that the unprecedented JWST data products are of the highest quality possible and available in the archive as quickly as possible. We also describe the rehearsals that are taking place, in order to ensure the operations systems, personnel, and procedures are ready well in advance of the spacecraft launch.

1. JWST Science and Operations Center Interfaces

The Data Management Subsystem (DMS) is one of six subsystems within the Science and Operations Center (S&OC), which serves as an element of the Ground Segment (see Figure 1). The DMS has interfaces with all of the other S&OC subsystems, as well as external interfaces to JWST Users and the International Data Centers. The DMS's main interface is with the Flight Operations Subsystem (FOS) which provides the science and engineering data received from JWST.

2. JWST Data Management Subsystem Components

The Data Management Subsystem is comprised of 14 components (see Figure 2), which together perform science and engineering data receipt, processing, archiving, and distribution functionality. Inputs from the Flight Operations Subsystem (FOS) and Proposal Planning Subsystem (PPS) are combined by the DMS to generate data products. These data products are then stored and made available for distribution to end users. The DMS is also instrumental in generating, storing and distributing data for the Wave Front Sensing and Control Software Subsystem (WSS), which is used for mirror alignment and focusing.

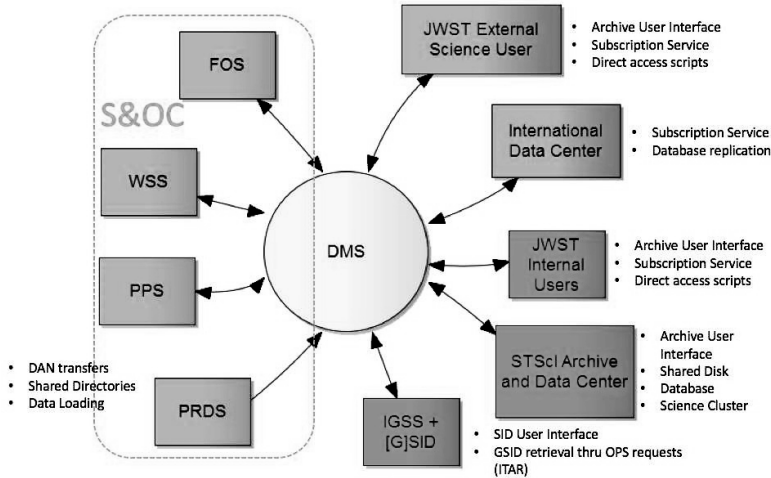


Figure 1. JWST Science & Operations Center Interfaces (Kyprianou & Alexov 2016)

3. JWST DMS Operations: Processing Control

The Open Workflow Layer (OWL) / HTCondorTM (HTCondor 2018) workflow manager controls the flow of new science and engineering data, reprocessed science data, Observatory Status Files (used for processing science data), and spacecraft ephemerides into the data processing pipelines. The OWL augments HTCondorTM with additional capabilities that facilitate tracking the data. Using the OWL GUI, operations staff can easily monitor and control the data processing workflow, and rescue failed workflow steps when necessary.

4. JWST Simulator Rehearsals

In order to exercise the end-to-end flow of data from JWST to the MAST archive, JWST DMS Operations has participated in a series of rehearsals using the JWST Observatory Test Bed (OTB) simulator.

4.1. Rehearsal Objectives

Each rehearsal serves a particular purpose, to ensure that the major activities necessary for JWST to be successful have been practiced to perfection. The primary goal of the WaveFront Exercises is to rehearse one of the most important activities and tightest constraints on the S&OC as a whole: the Wavefront Sensing & Control (WFS&C) activities necessary to align the mirrors of the telescope. The Science Operations Rehearsals serve to prepare the Science Operations teams for launch, commissioning, and regular operations, exercise the JWST Ground System and its capabilities, to test Operational Procedures, and to identify new procedures that are needed. The Normal Operations Rehearsals use the JWST OTB simulator to practice what normal, daily op-

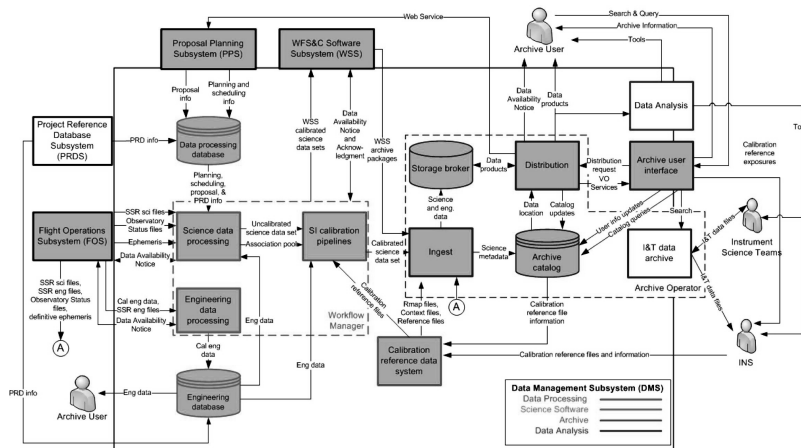


Figure 2. JWST DMS Components (Kyprianou & Alexov 2016)

eration of JWST on a regular (~monthly) cadence, to exercise the data modes necessary to meet the specific needs of daily science operations, and to support ongoing software development.

4.2. Rehearsal Successes

DMS has participated in multiple rehearsals to date, learning from each one and building on the successes and failures of previous tests. Each rehearsal showed improvement in the ability of all teams to identify, report, and resolve problems in real time. The DMS was able to meet the requirements to return WFS&C data products to the Wavefront Software Subsystem (WSS) within 90 minutes of receiving them from the spacecraft. An Calibrated Engineering data duplication issue was also identified, which was important to discover with sufficient time to investigate and resolve the issue. During the Science Operations Rehearsal #3, there were 21 DMS Operations procedures tested. Lastly, various communication channels were exercised, including Slack, Jabber, email, Voice Loop, and in-person communications. Important relationships were built, allowing for the teams to get to know people in other groups and their areas of expertise.

4.3. Rehearsal Lessons Learned

Many lessons were learned during the rehearsals that DMS has taken place in thus far. Most important of these lessons are the following:

- The OTB simulator configuration needs to be carefully curated to ensure that the versions of all systems installed are compatible with one another.
- It is important to include thorough regression testing of the operational string (~20 servers where data is received from the OTB simulator) to confirm proper configuration prior to each rehearsal.

- In order to ensure that resulting data products are as expected, a better method to predict what data products will be created from each observation program's Astronomer's Proposal Tool (APT) file is needed.

5. Summary

The Data Management Subsystem is a complex, multi-faceted system that performs the crucial role of receiving, processing, archiving, and distributing the data from JWST. It is one of 6 subsystems within Science and Operations Center element of the JWST Ground Segment. The DMS has interfaces with all of the other S&OC subsystems, with JWST Users, and with the International Data Centers. The Wavefront, Science Operations, and Normal Operations Rehearsals are integral to the success of JWST. These rehearsals serve to test the S&OC's ability to meet the mission's goals, and to provide practice for the teams involved, and to identify any issues or areas that need improvement before launch.

Acknowledgments. STScI is operated by the Association of Universities for Research in Astronomy, Inc. under NASA contract NAS 5-26555. We would like to thank NASA/ESA for funding the James Webb Space Telescope and the data archive at STScI. We would also like to thank the DMS Team and Team Leads, without whose expertise and dedication none of this would be possible.

References

- HTCondor 2018, Computing with htcondor website, <http://research.cs.wisc.edu/htcondor/>. Accessed: 2018-11-09
- Kyprianou, M., & Alexov, A. 2016, James Webb Space Telescope Mission Science and Operations Center Data Management Subsystem Design Document Revision A, Space Telescope Science Institute



Breakfast Buffet (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

CIAO: A Look Under the Hood of Chandra's X-Ray Imaging and Analysis Software Configuration Management – Past, Present, and Future.

Zeke Kaufman, Mark Czesitello-Dittmar, Janet D. Evans, Omar Laurino,
Warren McLaughlin, and Joe Miller

Smithsonian Astrophysical Observatory, Cambridge, MA, USA;
zeke.kaufman@cfa.harvard.edu

Abstract. The CIAO (Chandra Interactive Analysis of Observations) software suite is approaching two decades of service¹ and CIAO remains the primary analysis package from the Chandra X-Ray Observatory. Despite the package's maturity, CIAO continues to undergo active development from a diverse group of developers, using multiple programming languages and build infrastructures. Keeping up with the ever-evolving capabilities in hardware, software, version control systems, and paradigm shifts in software development methodologies presents a challenge to both developers and configuration management teams. This paper provides an overview of how the CIAO software suite has evolved over the years with a particular emphasis on configuration management of the system. Additionally, we describe CIAO's integration with various off the shelf software with a focus on recent changes with Python package management and distribution. We will conclude with an outlook on the future direction of CIAO infrastructure including possible integration with modern package management systems such as Conda and plans for Continuous Integration.

1. Overview

CIAO is the software package developed by the Chandra X-Ray Center for analyzing data from the Chandra X-ray Telescope. It can also be used with data from other Astronomical observatories, whether ground or space based. CIAO is actually a collection of tools that are available independently as stand-alone packages from the CIAO website², or can be downloaded together and installed as a single package. A brief description of the CIAO segments are as follows:

- **Core:** The core set of tools and libraries required for any segment of CIAO to properly run.
- **Tools:** A collection of imaging and analysis scripts and binaries.
- **Prism:** A graphical user interface (GUI) application for file browsing and simple editing.

¹CIAO version 1.0 was released October 1999

²<http://cxc.cfa.harvard.edu/ciao/>

- **ChIPS:** A plotting software package designed so that visualizations can be built up interactively and can easily be saved, printed, and restored. Visualizations may be created from FITS or ASCII format files, and include support for curves, histograms, contours, and images.
- **Sherpa:** The CIAO modeling and fitting application. It enables the user to construct complex models from simple definitions and fit those models to data, using a variety of statistics and optimization methods.
- **ObsVis:** A tool to aid in observation planning which allows the inspection of sky images with overlaid instrument fields-of-view (FoV).
- **CalDB:** Contains all the calibration files required for Chandra data analysis.
- **Contrib:** User contributed scripts and code to enhance the capabilities of CIAO.

2. Code and Version Control

CIAO is a partial subset of the much larger Data Systems (CXCDs) code base which supports proposal planning, mission planning, standard data processing, reprocessing, catalog processing, data archiving, data distribution/retrieval, and analysis for all data received from the Chandra X-Ray observatory.

The CXCDs code base is comprised of over 28,000 files approaching 2.2 million lines of code, stored in a ClearCase repository with a nearly 20 development year history. The CIAO subset of CXCDs contains roughly 5,000 files and several hundred thousand lines of code which is further divided into individual segments. It is the combined role of the configuration manager and developers to determine which files belong in CIAO versus code that strictly belongs in the CXCDs superset code base.

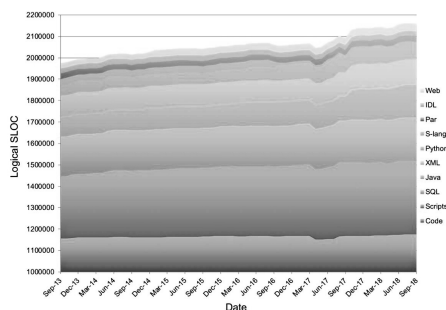


Figure 1. Logistical source lines of code of the CXCDs code base.

The ClearCase repository contains five of the eight CIAO segments (tools, chips, core, obsvis, prism). An online Github repository contains the Sherpa segment³. Sherpa is also available as a standalone git download (decoupled from CIAO). User contributed scripts are maintained by Science Data Systems (SDS) and CalDB files are maintained by the CalDB manager. These eight segments are inputs into the CIAO build infrastructure which is a collection of build scripts stored in a local git repository.

³<https://github.com/sherpa/sherpa>

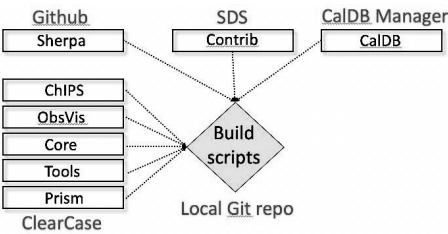


Figure 2. Version control interplay of the various CIAO segments.

3. Platform Support

CIAO 1.0 was released in 1999 and initially supported on Solaris and 32-bit Linux distros. Platform support history has evolved to meet user demand and a variety of Operating Systems over a 19-year development history and 20+ releases. The first release included command line tools and gui applications. Scripting capability via S-Lang was added in v2.0 (2000), and Python support was added in v4.0 (2007). The current build architecture for CIAO 4.11 involves building on three macOS machines (OSX 10.11, macOS 10.12, macOS 10.13) and two Linux boxes (CentOS 6.9 and Ubuntu 14.04 LTS). Smoke tests and regression tests are also executed on newer versions of these operating systems (as well as other Linux distros) to ensure platform compatibility.

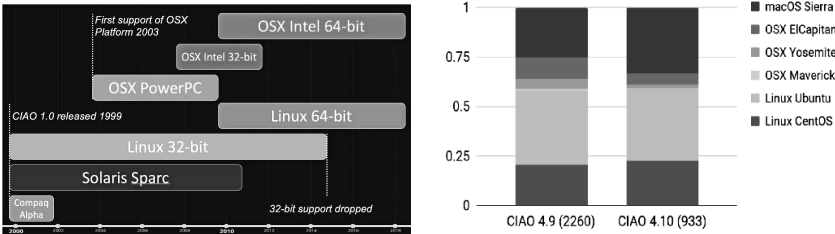


Figure 3. CDO surveys help determine supported platforms. Left: Timeline of CIAO platform support. Right: User download statistics of CIAO 4.9 and CIAO 4.10 releases.

4. OTS Software and Test Infrastructure

CIAO is mandated to be a self-contained package meaning users should be able to run the tools out-of-the-box without the need to install various third party libraries. Therefore all dependency packages are shipped which includes approximately 45 OTS and adapted OTS packages. Recent enhancements in the configuration management of the system involve the use and shipping of the Pip package manager within CIAO. Pip simplifies the configuration management of Python modules shipped with CIAO and also provides users the ability to integrate any of the wide array of python analysis modules into the CIAO software suite.

Configuration management of CIAO Python modules involves first creating a list of required core modules in a `requirements.txt` file. The Pip package manager will then determine all the dependencies for all supported platforms and retrieve them to a local Python module repository. This local repository is then used by the CIAO build infrastructure for installing Python modules. Storing the Python modules in a local repository rather than retrieving them from the Python Package Index (PyPI) provides for a tight and well controlled configuration management of the overall system.

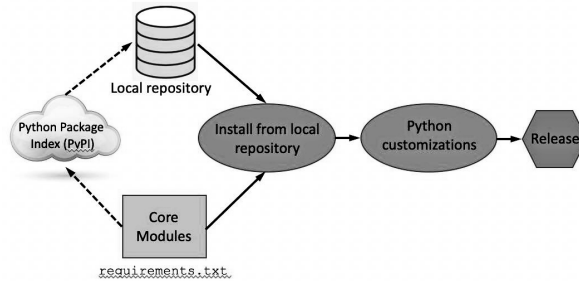


Figure 4. Python package management within CIAO involves populating a local Python repository from the online PiPY repo.

The CIAO build architecture is broken down into three separate builds running in parallel, *CIAOD*, *CIAOX*, and *CIAOT*. *CIAOD* builds are for development code. Development code that passes all tests get moved into the *CIAOX* builds reserved for integration code. Integration code undergoes more rigorous testing before being migrated in the *CIAOT* build which is the release build. CIAO will typically undergo three beta releases before the official release which usually occurs every December⁴.

5. Way Forward

Future directions for CIAO involve a simplified build infrastructure and new modes of distribution. We are currently investigating using Conda⁵ for internal OTS package management as well as a possible distribution model for CIAO. We are also exploring options for integrating the CIAO package into native package management systems `apt-get` and `yum`, simplifying the installation procedure for users on various Linux platforms. Additionally, now that CIAO build scripts are stored in a local git repository, we are investigating Configuration as Code practices to implement a continuous integration development cycle on the CIAO build framework.

⁴CIAO4.10 was released in April 2018 as opposed to December 2017

⁵<https://conda.io>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

DALiuGE/CASA Based Processing for the Extragalactic HI Observations with FAST

Vyacheslav Kitaeff,¹ Ming Zhu,² Lister Staveley-Smith,¹ Rodrigo Tobar,¹
 Kevin Vinsen,¹ Andreas Wicenec,¹ and Chen Wu¹

¹*The International Centre for Radio Astronomy Research,
 The University of Western Australia, Australia*

²*National Astronomical Observatories
 Chinese Academy of Sciences, Beijing, China*

Abstract. We present a prototype for the spectral-line data reduction pipeline based on the graph-based execution framework DALiuGE, and the CASA single-dish spectral-line package. The pipeline has been designed for the drift-scan mode of FAST multi-beam radio telescope targeting extra-galactic HI observations.

1. Introduction

The Five-hundred-meter Aperture Spherical radio Telescope (FAST) is planning a multi-beam multi-purpose survey that includes an extragalactic HI survey (Li et al. 2018).

We wanted to develop a pipeline that would be a flexible, scalable, and overall future-proof option to reduce FAST-HI extragalactic survey data. We started from investigating a few established options, including LiveData/Gridzilla, ASAP, proprietary code development, and CASA. We've selected the CASA (McMullin et al. 2007) single dish spectral line package that has been recently refurbished into a new more compact interface starting from version 5.0, along with the DALiuGE (Wu et al. 2017) execution framework. We have also integrated Next Generation Archive System (NGAS) software for data management purposes (Wu et al. 2013).

2. Rational for the selection of software packages

DALiuGE: The Data Activated Liu Graph Engine (DALiuGE) developed by ICRAR is an execution framework for processing large astronomical datasets at a scale required by the SKA1 (Wu et al. 2017), (<https://github.com/ICRAR/daliuge>). It includes an interface for expressing complex data reduction pipelines consisting of both data sets and algorithmic components and an implementation run-time to execute such pipelines on distributed resources. By mapping the logical view of a pipeline to its physical realization, DALiuGE separates the concerns of multiple stakeholders, allowing them to collectively optimize large-scale data processing solutions in a coherent manner. The execution in DALiuGE is data-activated, where each individual data item autonomously triggers the processing on itself. Such decentralization also makes the execution framework scalable and flexible.

NGAS: The Next Generation Archive System (NGAS) is a feature rich, archive handling and management system (<https://github.com/ICRAR/ngas>). In its core it is a HTTP based object storage system. It can be deployed on single small servers, or in globally distributed clusters. It is possible to run more than one server on a single host and it is possible to run many servers across hundreds of nodes as well as across various sites. It also allows mirroring the sites running independent NGAS clusters or running multiple clusters against a single database. The data can be archived and retrieved programmatically with data integrity being checked via various checksum methods. NGAS has a high customization via user-provided plug-ins. The standard distribution of DALiUGe included NGAS drop.

CASA: Although, CASA had been developed with the primary goal of supporting the data post-processing needs of the next generation of radio astronomical telescopes such as ALMA and VLA (McMullin et al. 2007), (https://casaguides.nrao.edu/index.php/Main_Page), the package can process both interferometric and single dish data. The CASA infrastructure consists of a set of C++ tools bundled together under an iPython interface as a set of data reduction tasks. This structure provides flexibility to process the data via task interface or as a Python script as a module in DALiUGe. In addition to the data reduction tasks, many post-processing tools are available for even more flexibility and special purpose reduction needs. The Single Dish tool was initially developed by CSIRO based on the ASAP software package. Beginning from release 5.0.0 the development is driven by ALMA. *Sdcal* function contains calibration modes that make CASA suitable tool for the planned drift-scan HI survey with FAST.

3. FAST-HI pipeline

The prototyped pipeline provides six modules:

1. *FASTcal* – based on *sdcal* that implements a single-dish data calibration scheme similar to that of interferometry, i.e., generate calibration tables (caltables) and apply them.
2. *FASTimaging* – mapping Tsys and Tsky calibrated data onto an image grid.
3. *FASTflagging* – based on *dataflag* that flags an MS or a calibration table.
4. *FASTMLflagging* – convolutional neural network inference automatic RFI flagging.
5. *FASTbaseline* – based on *sdbaseline* that performs baseline fitting/subtraction for single-dish spectra.
6. *FASTexportfits* – converts a CASA image to a FITS file in accordance with FITS 3.0 standard.

Each processing module can have any number of configuration files. If a module is executed without specifying one, it will try using a default configuration. If it cannot find the default configuration, it will create one. The *conf* directory contains some configuration files. These files are not the default configurations, but rather, those that were used during testing on ICRAR test system, and likely would need to be modified when the tests are done on a different system using different datasets.

There are three processing modes currently available and prototyped in DALiuGE logical graphs: real-time calibration, imaging, and reprocessing.

The real-time calibration pipeline assumes that the observation dataset is copied into NGAS archive as soon as it becomes available at the telescope. This will trigger deployment of `calibrate_pipeline` that will calibrate the observation and copy the resulting datasets into an archive.

The imaging pipeline provides gridding and imaging for selected observations. The configuration file contains a list of observations to be imaged. The image is produced as a measurement set and then exported as FITS cube.

The reprocessing pipeline combines flagging, calibration and imaging step from the archive. This is computationally extensive mode that normally requires a compute cluster.

CASA dataflag provides the algorithms to flag RFI. However, as part of the design we've included an option to use flagging based convolutional neural networks algorithm using PyTorch as a drop in DALiuGE. This technique is promising in characterization of the RFIs that are specific to the telescope in a location, therefore can be more accurate than signal processing based techniques in flagging difficult cases of RFI that have a broadband continues nature of the signal.

4. Summary

We developed a prototype of the data reduction pipeline for the planned FAST HI extragalactic survey that is scalable, extendable, and simple to use and develop further. Commissioning the FAST telescope with 19 beam receiver will allow testing the software on real data, and training machine learning based RFI flagging in near future.

References

- Li, D., Wang, P., Qian, L., Krco, M., Dunning, A., Jiang, P., Yue, Y., Jin, C., Zhu, Y., Pan, Z., & Nan, R. 2018, *IEEE Microwave Magazine*, 19, 112
- McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in *ADASS XVI*, edited by R. A. Shaw, F. Hill, & D. J. Bell, vol. 376 of *ASP Conf. Series*, 127
- Wu, C., Tobar, R., Vinsen, K., Wicenc, A., Pallot, D., Lao, B., Wang, R., An, T., Boulton, M., Cooper, I., Dodson, R., Dolensky, M., Mei, Y., & Wang, F. 2017, *Astronomy and Computing*, 20, 1. URL <http://www.sciencedirect.com/science/article/pii/S2213133716301214>
- Wu, C., Wicenc, A., Pallot, D., & Checcucci, A. 2013, *Experimental Astronomy*, 36, 679. 1308.6083



Dutch bribes (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Running GTC Data Reduction Pipelines in Jupyter

Sergio Pascual, Nicolás Cardiel, Cristina Cabello, Mario Chamorro-Cazorla,
 Cristina Catalán-Torrecilla, Bililign T. Dullo, África Castillo-Morales,
 Armando Gil de Paz, and Jesús Gallego

Universidad Complutense de Madrid, Madrid, Madrid, Spain;
sergiopr@fis.ucm.es

Abstract. The data reduction pipelines for the Gran Telescopio Canarias instruments EMIR and MEGARA are based on the same Python-based framework: numina. The instrumental pipelines can be run either automatically at the telescope site or using a command line interface. We have added support to run the pipelines inside a Jupyter notebook, with the Python kernel. The new classes in numina provide persistent storage of reductions and querying capabilities, to retrieve previous reductions.

1. Introduction

The data reduction pipelines for the GTC (Gran Telescopio Canarias) instruments EMIR (Garzón et al. 2016) and MEGARA (Gil de Paz et al. 2016) are based on the same framework: numina (Pascual et al. 2018).

Numina was designed with a modular architecture. Instrumental pipelines are plugins for the main numina module. Numina package provides tools to build DRPs as well as the API calls to interface with the GTC control system.

Furthermore, the I/O facilities (data serialization) are also modular. When run at the telescope, the reduction results are stored in the operation database, whereas in standalone mode, results are handled by numina and stored to disk. Thus, numina-based pipelines can be run either as a component of the GTC controls system or standalone.

The pipelines for both EMIR, PyEmir (Pascual et al. 2010, 2019) and MEGARA, megaradrp (Pascual et al. 2013) are currently deployed at the telescope. In the future, the pipeline of FRIDA (López et al. 2014) will be also based on numina.

2. Jupyter Notebooks

Jupyter (Kluyver et al. 2016) is web application that allows creating documents with live code, visualizations, equations and text. By means of different kernels it can run Python, Julia and R code. Jupyter notebooks are a great tools for data exploration and self-contained research.

We have added support to run the pipeline inside a Jupyter notebook, with the python kernel.

We have created a new API to expose to the user some of the capabilities that the command-line interface of numina has internally; such as persistent storage of reductions and querying capabilities, to retrieve older reductions

Given the modular design of numina, the different pipelines do not need any modification to work inside Jupyter.

The following is a runing example, based on MEGARA IFU images. Figure 1 shows the initialization of the pipeline. Recipes log most of their activity. In a notebook, logging appears in the output cell. In the first cell we restrict logging to INFO to avoid too much text. In the middle cell we create a DataStore object. It is a database-like container that keeps track of reductions and products. The datastore can be written back to disk if needed. Observation files contain the names of raw files and the observing mode in YAML format. We load them inside the data store (bottom cell).

```
from numina.user.helpers import init_datastore_file
import numina.user.baserun as base
import logging
logging.basicConfig(format='%(levelname)s: %(message)s', level=logging.INFO)
```

```
print(numina.__version__)
```

```
0.18.dev0
```

```
datastore = init_datastore_file(controlfile='control2_base.yaml')
```

```
INFO:reading control from control2_base.yaml
INFO:control format version 2
```

```
obfiles = ["0_bias.yaml", "1_tracemap.yaml", "2_modelmap.yaml", "3_wavecali
"5_twilight.yaml",
"6_Lcbadquisition.yaml",
"7_Standardstar.yaml",
"8_reduce_LCB.yaml"]
```

```
# all obs = []
for obfile in obfiles:
    u = datastore.load_observations(obfile)
```

Figure 1. Logging initialization (top cell). Creation of DataStore object (middle cell). Observation files loaded in data store (bottom cell)

Once the observations are loaded we can command new reductions. In Figure 2 we show the begining of the output of the reduitiion of an observing block in LCB reduction mode. The datastore keeps track of calibrations and the results of previous reductions. The `run_reduce` method queries the datastore to load the correct calibration set.

The result of the reductions can be manipulated inside the notebook. The result of the reduction is stored in the attribute `.result` of the return value of `run_reduce` (task8 in Figure 2).

In Figure 3 we are accessing the results of the previous reduction, with name `reduced_rss`. The object is a standard `fits.HDUList` from `astropy`.


```
task8 = base.run_reduce(datastore, "8_HR-R")
INFO:processing OB with id=8 HR-R
INFO:search master_dark of type MasterDark()
INFO:type MasterDark compatible with tags {'vph': 'HR-R', 'insmode': 'LCB'}
not found
INFO:search master_bpm of type MasterBPM()
INFO:found /home/spr/Documents/Congresos/GH2018/MEGARA/ca3558e3-e50d-4bbc-86bd-da50a0998a48/MasterBPM/master_bpm.fits
INFO:search master_slitflat of type MasterSlitFlat()
INFO:type MasterSlitFlat compatible with tags {'vph': 'HR-R', 'insmode': 'LCB'}
not found
INFO:search master_traces of type ModelMap()
INFO:type ModelMap compatible with tags {'vph': 'HR-R', 'insmode': 'LCB'}
not found
INFO:running recipe
INFO:starting LCB reduction
INFO:loading BPM
```

Figure 2. With the observation loaded, we can command the reductions using the id field of the observations. The datastore keeps track of previous reductions and calibrations.

```
rss = task8.result.reduced_rss.open()

plt.imshow(rss[0].data)

<matplotlib.image.AxesImage at 0x7f021693e400>
```

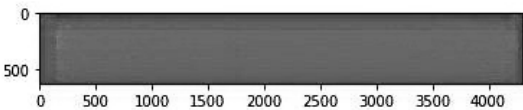


Figure 3. Showing the 623 spectra from MEGARA

In Figure 4 we show a possible manipulation of the result of reduction task. The processed image is loaded directly from the result of the reduction. In particular, we cut a slice of the reduced image and then we visualize it with the help of the custom hexagonal plotting routine in the MEGARA pipeline hexplot.

Acknowledgments. This work was funded by the Spanish Programa Nacional de Astronomía y Astrofísica under grant AYA2016-75808-R, which is partially funded by the European Development Fund (ERDF).

References

- Garzón, F., et al. 2016, in Ground-based and Airborne Instrumentation for Astronomy VI, vol. 9908 of Proceedings of SPIE, 99081J
- Gil de Paz, A., et al. 2016, in Ground-based and Airborne Instrumentation for Astronomy VI, vol. 9908 of Proceedings of SPIE, 99081K


```

from astropy.wcs import WCS
wcs = WCS(rss['FIBERS'].header)

rssdata = np.squeeze(rss[0].data[:, 2000])
zdisp = rssdata

fig = plt.figure()
ax = fig.add_axes([0.15, 0.15, 0.8, 0.8], projection=wcs)
scale = 0.443
ax.coords.grid()
ax.set_xlim([-5.5, 5.5])
ax.set_ylim([-5.5, 5.5])
col = vis.hexplot(ax, x, y, zdisp, scale=scale)
plt.show()

```

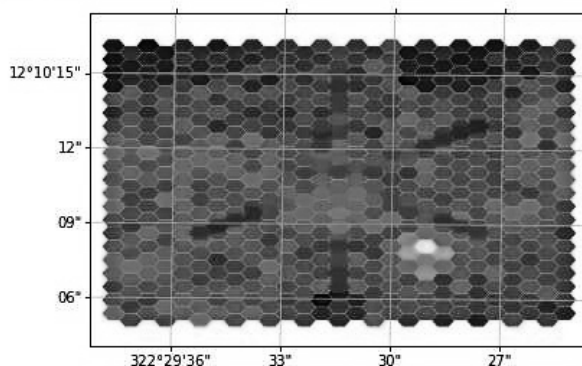


Figure 4. Visualization of a slice of a processed MEGARA image

- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. 2016, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides, & B. Schmidt (IOS Press), 87
- López, J. A., et al. 2014, in *Ground-based and Airborne Instrumentation for Astronomy V*, vol. 9147 of ASP Conf. Ser., 91471P
- Pascual, S., Cardiel, N., & Molgó, J. 2019, in *ADASS XXVI*, edited by M. Molinaro, K. Shortridge, & P. Pasian (San Francisco: ASP), vol. 521 of ASP Conf. Ser., 232
- Pascual, S., Cardiel, N., & Picazo, P. 2018
- Pascual, S., Eliche-Moral, M. C., Villar, V., Castillo, Á., Gruel, N., Cardiel, N., Carrasco, E., Gallego, J., Gil de Paz, A., Sánchez-Moreno, F. M., & Vílchez, J. M. 2013, in *Astronomical Data Analysis Software and Systems XXII*, edited by D. N. Friedel, vol. 475 of *Astronomical Society of the Pacific Conference Series*, 287
- Pascual, S., Gallego, J., Cardiel, N., & Eliche-Moral, M. C. 2010, in *ADASS XIX*, edited by Y. Mizumoto, K.-I. Morita, & M. Ohishi (San Francisco: ASP), vol. 434 of ASP Conf. Ser., 353

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Reprocessing All the XMM-Newton Scientific Data: A Challenge for the Pipeline Processing System

José-Vicente Perea-Calderón¹, Pedro Rodríguez-Pascual², and Carlos Gabriel²

¹*RHEA for ESA/ESAC, European Space Astronomy Center (ESAC-ESA), Madrid, Spain; jose.perea@sciops.esa.int*

²*XMM-Newton SOC, European Space Astronomy Center (ESAC-ESA)*

Abstract. 2019 will mark the 20-year anniversary of the XMM-Newton Mission¹. So far, the mission has successfully completed a total of around 14000 pointing observations, and it is expected to continue for many more years, producing a huge number of high-quality science data products.

Data processing of those observations is carried out by the XMM-Newton Pipeline Processing System (PPS)² and the products are delivered to the XMM-Newton Science Archive (XSA)³. During the two decades many changes have been implemented in the data processing software, partly following improvements to the calibration of the science instruments. Several re-processing campaigns have been undertaken along the mission in order to have an up-to-date and uniformly processed set of high-level science data products in the archive.

This paper is a review of the analysis that has been carried out to achieve re-processing of the science data of the whole mission, and to find a more effective way to do it in the future.

1. Introduction

The XMM-Newton mission re-processing campaign requires a power computer infrastructure to be done. But even so the overall processing time could be considered too long depending on the number of observations, the complexity of the software algorithms and the calibration, etc. Substantial reduction of the data processing time would allow change of the frequency of the renewal of the archive contents.

Unlike the daily mission operations where a limited number of observations have to be processed by PPS, a whole mission re-processing is a real challenge. An individual XMM-Newton Pipeline job (processing one observation) can take several hours of computer processing time. To achieve the processing of thousands of observations in a reasonable period of time requires a special preparation including a deep analysis of the computing resources. An extreme optimization of the resources sharing becomes essential in this case.

¹XMM-Newton SOC home page <http://xmm.esac.esa.int>.

²XMM-Newton Pipeline <https://www.cosmos.esa.int/web/xmm-newton/pipeline>.

³XMM-Newton Science Archive <http://nxsa.esac.esa.int>.

Besides the optimization of the computing infrastructure usage, a set of software tools had to be developed in order to cope with the management and monitoring of this enormous number of individual Pipeline jobs.

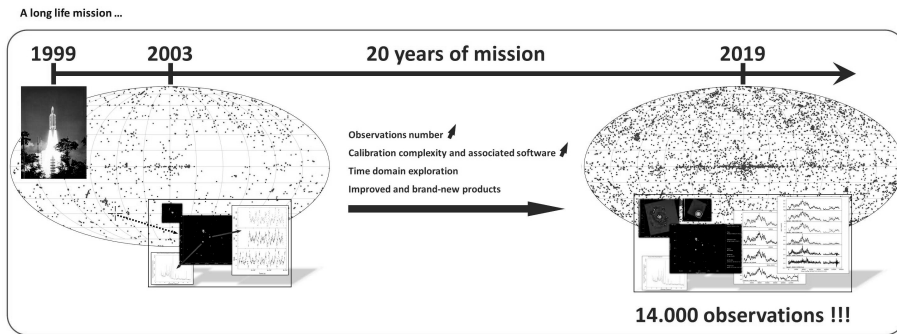


Figure 1. Increase of complexity over the years

2. All XMM-Newton Mission Re-processing

In addition to the increase of the number of observations over the years, the complexity of the calibration and the associated software has also increased significantly. As a result the Pipeline produces improved and brand-new data products. It is also expected to provide new results from the time domain exploration of the data: variability, transients detection and others.

The considerable improvements in the quantity and quality of the results have resulted in subsequent increased workload in the computer infrastructure and, as a consequence, an important increase of the processing time.

The real challenge is not processing all the observations of the mission itself but finding out how fast we can do the job. Processing all the mission in a short period of time would allow population of the archive on a continuing progressive and dynamic basis. So any significant change in the calibration or important software upgrade would produce a new “all-mission pipeline products pack” ready to be ingested in the archive.

In order to put this idea in place we have set ourselves the goal of processing the whole mission within a week.

3. XMM-Newton Pipeline Single-job

The first approach to speed up the process is splitting every Pipeline job into 4 threads which provides a process time reduction by a factor 2. But still, most of the Pipeline jobs may take up to 6 hours to complete in any of the computer nodes of the ESAC/ESA Grid infrastructure. In addition, many of those jobs might need up to 8 GB of memory to be processed (Figure 2).

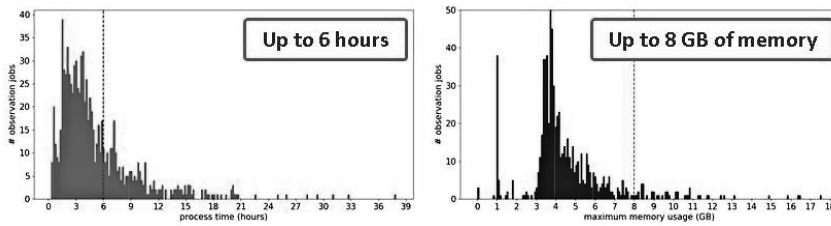


Figure 2. Distribution of the processing time per single-job (red plot). Distribution of the memory usage per single-job (blue plot)

4. Processing Computer Infrastructure

The setup of the XMM-Newton Pipeline processing system is settled into the ESAC Grid computer infrastructure. Pipeline jobs are therefore under the Grid computing model rules. One has to request in advance a particular amount of computing resources in order to get them guaranteed. For example, if a Grid job tries to use more memory than requested the job is killed to avoid memory overflow. Conservatively so far, each Pipeline job has been initially set up to request a significant amount of memory to avoid that jobs were killed by the Grid rules. The CPU resource, moreover, has been requested based on the parallelization Pipeline approach, where each job is split into four threads. Four CPU cores are reserved for every pipeline job. This setup limits the number of jobs that may run simultaneously in the Grid.

This initial setup has been modified in a very intensive campaign of high workload computing in order to figure out the real limitations of the systems, the setup and the infrastructure. After that we got two significant findings:

- The maximum demanded memory by each single job only happens in very short time peaks, so there is a lot of free memory most of the processing time. This indicates that any memory usage peak is absorbed by the system with no overflow risk.
- When many simultaneous jobs are running on the Grid the total CPU's load hardly reach 50 % of the total infrastructure CPU power

This memory and CPU usage behavior allows us to go further in terms of resources sharing on our computer infrastructure. Therefore the system has been forced to reach almost 100 % of the CPU load capacity and the jobs are launched with no memory restriction. By working in this way, we are able to have 140% more simultaneous jobs than in the initial conservative setup, and without losing processing speed of every single job. The whole Grid infrastructure is able to absorb all of this processing workload.

Thanks to this new Pipeline setup the overall processing time of the all-mission re-processing job will be significantly reduced. In the absence of final results, we estimate through extrapolations that this huge and intensive work will take less than one week.

5. Conclusions

- We can reprocess all the mission at any time, and then populate the complete XMM-Newton Science Archive with the best science data products at any time as well.
- Requirements for every single-job matter. Each single-job demands a specific amount of computer resources, and therefore the knowledge of that allows sharing the resources in a more effective way.
- Deep analysis of the computer capabilities is essential to accomplish the process of this huge number of jobs within a reasonable time.
- New monitoring and management tools become a necessity. Unlike the daily routine Pipeline data processing where a small number of jobs are managed, a bulk re-processing of the mission requires the development of reliable and robust tools to carry it out.



Current POC chair Nuria Lorente with past POC chair and future retiree Carlos Gabriel, final goodbyes at Thursday's ADASS box lunch (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

MeerKAT: Operational Workflow and Data Analysis

Rosly Renil

SARAO/NRF, Cape Town, Western Cape, South Africa; rosly@ska.ac.za

Abstract. MeerKAT, the next generation radio telescope is already operational for commissioning and analyzing large datasets for science community. This poster explains the operational processes in place for engineers, scientists, commissioners, etc to analyze the data-sets for further data mining that improves various processes. It is also worth noting the software processes and hardware involved, different analysis tools used for telescopic operations in producing good science data.

1. Introduction

The MeerKAT radio telescope is taking a new shape in the South African Karoo Radio Astronomy Reserve. MeerKAT is a precursor to the Square Kilometer Array (SKA) for the mid-frequency dish array. This world-class radio telescope will be capable of doing transformational science and will be the largest and most sensitive radio telescope array in the southern hemisphere until it is surpassed by the SKA.

The MeerKAT array consists of 64 dishes, a 13.5 m projected diameter dish each with an offset Gregorian configuration, giving it a sensitivity of between 300 and 400 mJ/K in frequency range 0.9 to 1.67 GHz. An offset optical configuration has been chosen because its unblocked aperture provides uncompromising optical performance and sensitivity, excellent imaging quality, and good rejection of unwanted radio frequency interference from satellites and terrestrial transmitters. It also enables the installation of multiple receiver systems in the primary and secondary focal areas, and provides a number of operational advantages.

This paper describes Science Data Processing teams software processes deployed on the telescope system. It also gives insight to various user interface tools used for health monitoring of different hardware and software systems running on-site.

2. MeerKAT Science Data Processing Process Flow

A dedicated and systematic process flow depicted in Figure 1 below explains the software process running on an automated testing system via Jenkins continuous integration system.

The highlights of Software Development Life Cycle (SDLC) below:

1. Proper management of software processes and tools.
2. Automated Jenkins build system run with every code change made in the repository.

- 3. Tests run internally on lab systems before deploying to production systems.
- 4. Efficient deployment of software code across different servers at the same time.
- 5. Docker registry and dockerised containers in managing different running versions.
- 6. UI tools in monitoring the overall health of the telescope system followed by access to logs, graphs and various signal-path chain activities.
- 7. UI admin related tools for monitoring the health of different hardware systems deployed on site and also monitoring systems at CHPC.

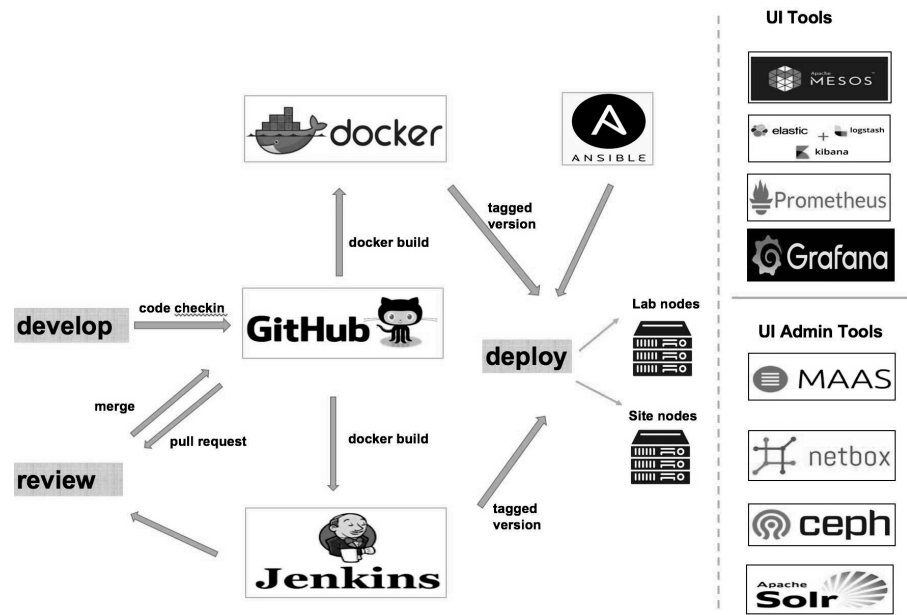


Figure 1. SDLC Workflow.

3. MeerKAT Science Operations and Data Rates

The MeerKAT telescope commissioned in July 2018 is already producing science results with the first early MeerKAT science image of the milky way center released. The telescope is fully operational during the week for various engineering debug and maintenance activities onsite. Dedicated science observations are scheduled and run for more than 12 hours in the evenings and long weekends.

The data rate currently coming out of the telescope is 2 Tb per second with large data file sets produced and ingested into the MeerKAT archive. The MeerKAT data link from Karoo site to Cape Town is soon to be upgraded to be 100 Gb per second; which will help in faster transfer and downloading of data-sets. See Figure 2 below.

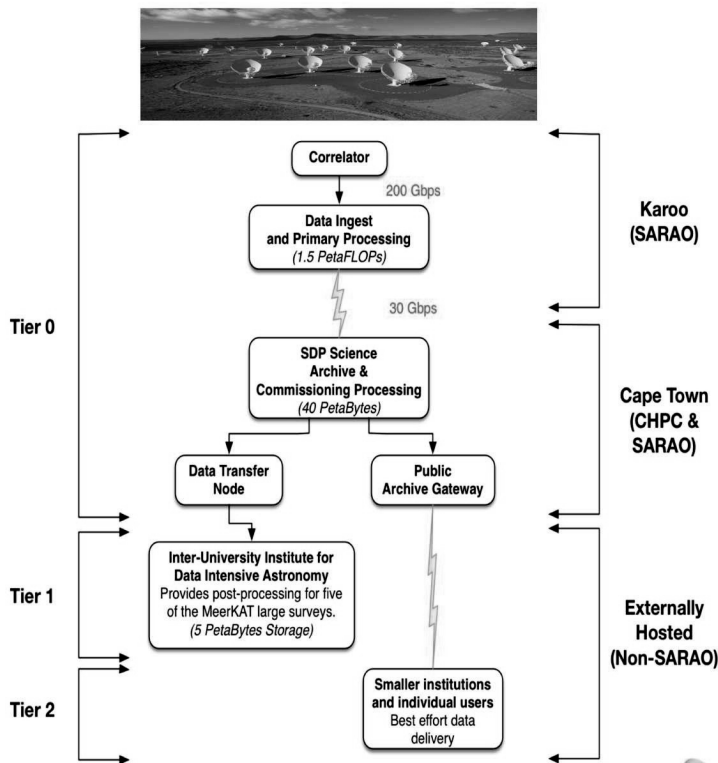


Figure 2. MeerKAT Data Rates.

4. MeerKAT Archive and Data Storage

The MeerKAT archive has 55 storage pods deployed at the Center for High Performance Computing (CHPC), Cape Town. This Peralex in-house built high capacity storage node solution uses Ceph cluster that helps in better archiving and processing of large data-sets for scientific user community.

The first cluster hosted at CHPC has 12 storage pods, which has 5.4 PiB of storage. The rest of the storage pods are hosted on CHPCs second cluster, which has a little over 12 PiB of storage. The composition consists of 35 Peralex high capacity storage nodes, 1680 8TB spinning disks and 35 NVMeS. Eventually these two clusters, will all be merged into one cluster of 20 PiB.

In terms of science data rates, Science Data Processing team is busy developing the full telescope operating mode rate (32K correlator mode) into the archive which will be around 20 Gbps. Testing is under progress, with full functionality expected in first quarter of 2019. There are 5 storage pods at the Karoo site used for the link buffer and other site services which comes to about 2 PiB. Besides, there is also the StorageTek SL150 tape library in Cape Town which has 3000 x 6TB tapes in it. See Figure 3 below.

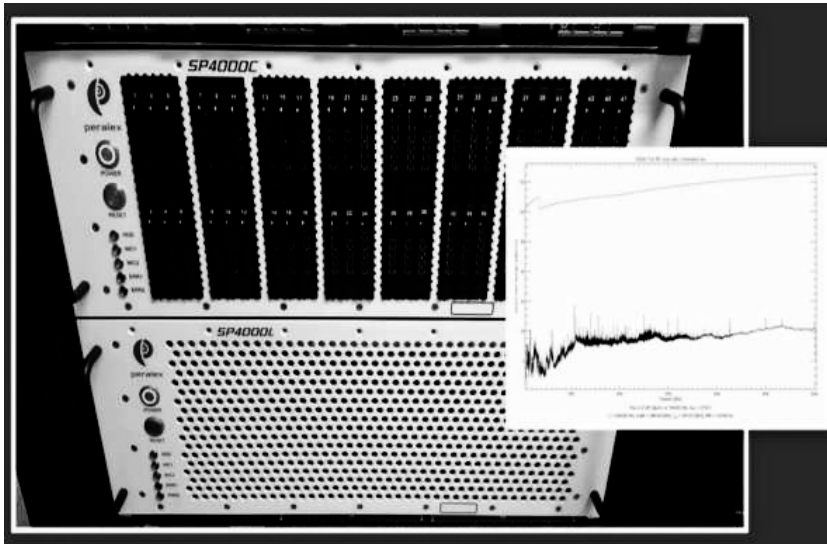


Figure 3. MeerKAT Storage Pods.

5. MeerKAT Data Analysis Tools

MeerKAT Science Data Processing Team incorporates a variety of visual analytics tools that is used at telescope operations level and for trouble-shooting.

1. Kibana, runs on top of Elasticsearch and is used primarily for analyzing log messages.
2. Grafana used for analyzing and visualizing metrics such as system CPU, memory, disk and I/O utilization.
3. Automated Jupyter/iPython notebooks run daily for generating user-level observation reports.
4. Exploring various Machine Learning techniques and tools

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Euclidizing External Tools: An Example from SDC-IT on How to Handle Software and Humanware

Erik Romelli,¹ Marco Frailis,¹ Samuele Galeotta,¹ Daniele Tavagnacco,¹
Davide Maino,^{2,3} Claudio Vuerli,¹ Gianmarco Maggio,¹ and Giuliano Taffoni,¹
ON BEHALF OF THE EUCLID CONSORTIUM

¹*INAF, Osservatorio Astronomico di Trieste, Trieste, Italy*

²*Università degli Studi di Milano, Milano, Italy*

³*INFN, National Institute for Nuclear Physics, Milano, Italy*

Abstract. Euclid is an upcoming space mission aimed at studying the dark Universe and understanding the nature of the so called Dark Matter and Dark Energy. Overall, Euclid will produce about 30 Petabytes of image data and the data processing will be a crucial aspect of the mission. A distributed computing system, with resources located in several Science Data Centers (SDCs), has been implemented and any software designed for Euclid must comply with a predefined framework. SDCs are in charge of the integration of external code within the official Euclid software environment. We will present an overview of the work performed by the Italian SDC in Trieste, taking into account the technicality and the crucial, but usually ignored, aspect of human interfaces.

1. The Euclid Mission

Euclid is a mission selected by the European Space Agency (ESA) at the end of 2011 to understand the nature of the dark Universe. Observations conducted on the Cosmic Microwave Background Radiation (CMB) proved the existence in our Universe of two dominant components whose nature is entirely unknown (Planck Collaboration 2015). Of the energy density of the Universe, 68.3% is in the form of Dark Energy (DE), which is causing the Universe expansion to accelerate, while another 26.8% is in the form of Dark Matter (DM), which exerts a gravitational attraction as normal matter, but does not emit or absorb light. The Euclid mission will investigate the distance-redshift relation and the evolution of cosmic structures by means of two instruments: the Visual Imager (VIS) and the Near-Infrared Spectrometer and Photometer (NISF).

2. Euclid Science Ground Segment

The Euclid ground segment consists of two blocks: the Operations Ground Segment (OGS), covering the mission control components and managed entirely by ESA, and the Science Ground Segment (SGS), for which the management is shared between ESA and the Euclid Consortium (EC).

The SGS, whose schematic representation is shown in Figure 1, is responsible for data processing and archiving and its mainly composed by the Science Operations

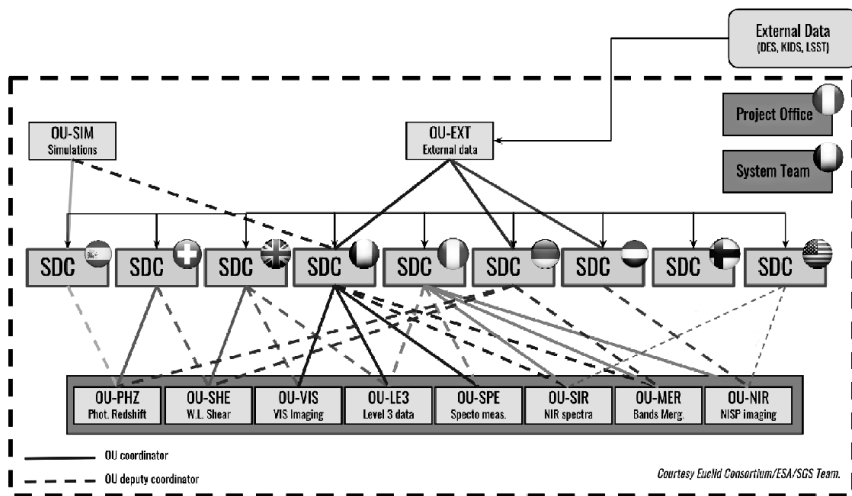


Figure 1. Schema of the Euclid Science Ground Segment.

Center (SOC), operated by ESA, and nine Science Data Centers (SDCs), located in Finland, France, Germany, Italy, the Netherlands, Spain, Switzerland, United Kingdom and United States. SDCs are computing centers in charge of instrument-related processing, production of science data products, simulations, ingestion of external data and in general all science-driven data processing. They will furthermore support the computational needs of the Instrument Operation Teams (IOTs). One of the duties of the SDC is to integrate external software components in the reference Euclid software environment.

SDCs work in synergy with working groups called Organization Units (OUs). Each OU is responsible for a different aspect of the scientific data processing:

- SIM: realizes the simulations needed to test, validate, and qualify the whole pipeline;
- VIS, NIR, SIR : are, respectively, in charge of processing the visible imaging data, the near-infrared imaging data and the near-infrared spectral data;
- EXT: in charge of ingesting in the pipeline all external data;
- MER: realizes the merging of all the calibrated data;
- SPE: extracts spectroscopic redshifts;
- PHZ: computes photometric redshifts from the multi-wavelength imaging data;
- SHE: computes shape measurements on the visible imaging data;
- LE3: computes the high-level science data products.

The Italian Science Data Center (SDC-IT) is based at the Astronomical Observatory of Trieste (OATs) and is the primary reference SDC for OU-NIR, OU-MER and OU-SIR, while it represents the auxiliary reference SDC for OU-SPE and OU-LE3.

3. Euclid Development Environment

The Euclid Development ENvironment (EDEN) is the common reference for standards, coding rules, tools and corresponding versions that must be applied to develop Euclid software. All the computing infrastructure, including both develop and production instances, must comply with this environment. In order to help the developers with the adoption of reference coding standards and libraries, this environment is implemented as a Local DEvelopment ENvironment (LODEEN), a virtual machine which implements a set of tools in addition to EDEN, such as quality tools, code generation scripts and version control. Continuous integration and delivery of the software are provided by a development support platform called COLlaborative DEvelopment ENvironment (CODEEN).

4. Highlights from the Italian Science Data Center

Not all the code is designed and implemented *ex novo* for Euclid purposes; usually data analysis pipelines inherit already existing software tools, designed outside the Euclid Consortium. If necessary and required by the OUs, SDCs are also in charge of porting and integrating external software modules in EDEN. As the primary reference SDC for OU-MER, SDC-IT team has ported the following external tools:

- ASTERIsM (Tramacere A. et al 2016): a collection of tools for detection and deblending of astronomical sources and extraction of morphometric features;
- A-Phot (Merlin E. et al 2018, submitted to A&A): a software designed for high precision aperture photometry;
- T-Phot (Merlin E. et al 2015) (Merlin E. et al 2016): a software designed for high precision, prior-based photometry in crowded, deep extra-galactic fields;

Furthermore, as an auxiliary SDC for OU-LE3, we did a significant work in terms of integration and optimization of algorithms needed to compute statistics related to galaxy clustering analysis. In the following section we present some highlights of the work done to integrate external software components in the Euclid software environment.

A software tool compliant with the EDEN environment must be written in C++11 or Python 3. External software is not necessarily constrained and can be written in one or more different languages. One of the activities of the SDC-IT development team was to migrate the photometry analysis (A-Phot and T-Phot) from the original C/C++/Python hybrid version to pure C++. For software written in Fortran 90, with a procedural paradigm, a complete object-oriented redesign and implementation in Python was also necessary.

Changes in software design are often required, since a Euclid compliant project has strict rules dealing with the source code structure and code quality metrics. Software maturity assessments are conducted in order to evaluate each software project architecture and quality. The architecture of the source deblending module implemented in ASTERIsM, for instance, has been modified in order to comply with EDEN coding rules. Moreover, the input and output interfaces of an external software components are modified and designed to implement the Euclid the Euclid Common Data Model,

a centralized repository defining all the SGS software interfaces, formalized with the XML Schema Definition language (XSD).

EDEN defines the version of software libraries that must be used in the development. This sometimes creates dependency issues: ASTERIsM, for instance, depends on more recent versions of Python modules available in EDEN. We modified the Euclid compliant version of the ASTERIsM deblending module to make it work with older APIs.

One of the tasks of SDCs is to optimize the code implemented in Euclid compliant projects. The EDEN compliant version of T-Phot has been significantly redesigned and optimized, improving both its RAM memory consumption, I/O efficiency and computing time.

5. Humanware

Humanware is an underrated but crucial aspect to take into account when dealing with a complex mission such as Euclid. The EC involves different people with different mindset and coming from different backgrounds: software engineers, for instance, are focused on the technical aspects while scientists care more about the reliability of the computed scientific outcome. Coding rules and a common environment help, but the hard part sometimes is to manage human resources, in particular human interfaces between SDC and OU teams and, in general, between the EC and the authors of external software tools, who are not necessarily constrained by Euclid regulation. A constant and constructive discussion is necessary to achieve the mission objectives: teleconferences, organized on a weekly or bi-weekly basis, help us to keep updated with the activities of the OUs referencing to SDC-IT. Face-to-face meetings with OU people are also a valuable tool, allowing a more direct interaction and representing in fact small workshops focused on solving the most pressing open issues.

Working with groups of people distributed in different institutes all around the world and heterogeneous in terms of expertise, a crucial aspect to be considered is the organization of human resources. In that sense, a key point of humanware handling, according to SDC-IT experience, is to fit each person in the right task, according to his/her field of competence, and trust each element of the team.

Acknowledgments. The authors acknowledge the Italian Space Agency (ASI), which supports the participation in Euclid of Italian institutions under the grant no. I/023/12/0.

References

- Merlin E. et al 2015, *Astronomy & Astrophysics*, 582, 21
- 2016, *Astronomy & Astrophysics*, 595, 7
- Planck Collaboration 2015, *Astronomy & Astrophysics*, 594, 38
- Tramacere A. et al 2016, *Monthly Notices of the Royal Astronomical Society*, 463, 2939

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

A Real-Time Data Reduction Pipeline for the Goodman Spectrograph

Simón Torres-Robledo,¹ and César Briceño^{1,2}

¹*SOAR Telescope, La Serena 1700000, Chile; storres@ctio.noao.edu*

²*Cerro Tololo Interamerican Observatory, Casilla 603, La Serena 1700000, Chile*

Abstract. The Goodman Spectroscopic Pipeline is reaching some maturity and behaving in a stable manner. Though its improvement continues, we have started a parallel effort to develop a real-time version, with the goal of obtaining fully reduced spectra seconds after the data has been obtained at the telescope. Most of the required structure, algorithms and processes already exist with the offline version. The real-time version differs in its requirements for flow control, calibration files, image combination, reprocessing, observing logging assistance, etc. Here we present an outline of the route for implementation of a real time online version of the Goodman spectroscopic pipeline.

1. Introduction

The 4-m Southern Astrophysical Research telescope *SOAR* telescope, located on Cerro Pachón, northern Chile, currently has as its most used instrument the *Goodman High Throughput Spectrograph* (GHTS), an imaging spectrograph developed by the University of North Carolina Clemens et al. (2004). *SOAR* operations are run in classical mode, with approved proposals scheduled on specific dates. Though observers can go up to the summit, most often they observe from a remote location via an Internet connection, through a VPN tunnel, accessing the instrument software with a VNC client. With the advent of the new generation of survey telescopes such as LSST, *SOAR* is aiming at becoming a prime follow up facility for transients and Time Domain events. To this effect, a project has been set up in collaboration with the National Optical Astronomical Observatory (NOAO) and Las Cumbres Observatory (LCO), to automate various processes in order to make observations more efficient, allow the telescope to respond faster to incoming alerts, and interface smoothly with the existing robotic scheduling technology developed by LCO.

The Goodman Spectroscopic Pipeline (GSP) is part of this effort, with an online version envisioned, working automatically and capable of delivering science-ready spectra seconds after the shutter has closed. This is particularly important when scientists need to make real time decisions like whether to obtain additional spectra of a given object that may be on the rise, or fading. However, at the moment of writing GSP exists only as an offline version, a one-line command that can be run once the observing night has finished. The user connects via VPN and VNC to a dedicated data reduction machine on which we have the latest version of the GSP. In this contribution we describe the requirements for an online implementation of the GSP using web tech-

nology with secure authentication and responsive layout, using modern, well proven technology.

2. Goodman Spectroscopic Pipeline: offline version status and performance

The GSP is based on a two-step process. `redccd` does the basic CCD data reduction, common for imaging and spectroscopy, with some small differences for spectroscopy. `redspec` does spectroscopic-specific data reduction Massey & Hanson (2013), including automatic wavelength calibration. Processing a typical full night takes between three and five minutes.

From the start of this project, we aimed for an automated, unsupervised, wavelength calibration of the spectra. After experimenting obtaining wavelength calibrations with several methods, including trying out an interactive module, we found that the best solution was to use a catalog of templates, a collection of comparison lamp spectra taken with our own hollow cathode lamps. This approach provides good, reliable solutions without the need for any manual interaction from the user. The GHTS is a highly configurable instrument, which means that the camera and grating angle can be adjusted to almost any combination of values, yielding a wide range of possible wavelength coverage options. This flexibility leads to practical problems when trying to setup a library of comparison lamps for various modes, because of the large number of possible wavelength ranges. Therefore, we decided to setup the standard spectral lamp library limited to the most often used modes, for the more frequently requested gratings. It is still possible to use the instrument in *Custom* mode (in which the user defines the central wavelength for the Littrow mode) but such custom modes do not have a corresponding template in the library, for obvious reasons. There are seven gratings available at present for the Goodman spectrograph: 400, 600, 930, 1200, 1800, 2100, and 2400 *l/mm*; for the last three there is no fixed mode defined, but rather they are normally used in Littrow mode. We built standard lamp spectra for the two most used modes of the 400 line grating, and for several modes of the 600, 930 and 1200 line gratings.

Table 1. Sample of spectroscopic modes definition for the 930 *l/mm* grating

Grating (lines/mm)	Dispersion (Å/pixel)	Coverage (Å)	Max R @ 5500nm (3 pix with 0.46" slit)	Blocking Filter
930	0.42	M1: 300-470	4450	—
		M2: 385-555		—
		M3: 470-640		GG-385
		M4: 555-725		GG-495
		M5: 640-810		GG-495
		M6: 725-895		OG-570

The lamps in the library are not linearized because the raw spectra coming out of the GHTS are non-linear in wavelength space, therefore reference wavelength solutions need to be non-linear. All the emission lines detected in the lamp spectrum are recorded in the header, together with their corresponding wavelength value as obtained from the fit of the mathematical model used to describe the solution.

Performance-wise the results have been quite satisfactory. Overall, we obtain root-mean square (RMS) values similar to those obtained using IRAF. For instance using the 930 l/mm grating in the M2 mode the RMS error of the wavelength solution was 0.281Å.

3. Goodman Spectroscopic Pipeline Real-Time version: Design Constraints

For the live data reduction pipeline we want to do away with the VNC system and move to a web-based service, that would allow secure authentication. Also, the offline version of the GSP exists as a single Python package; for the live version we will need several other components.

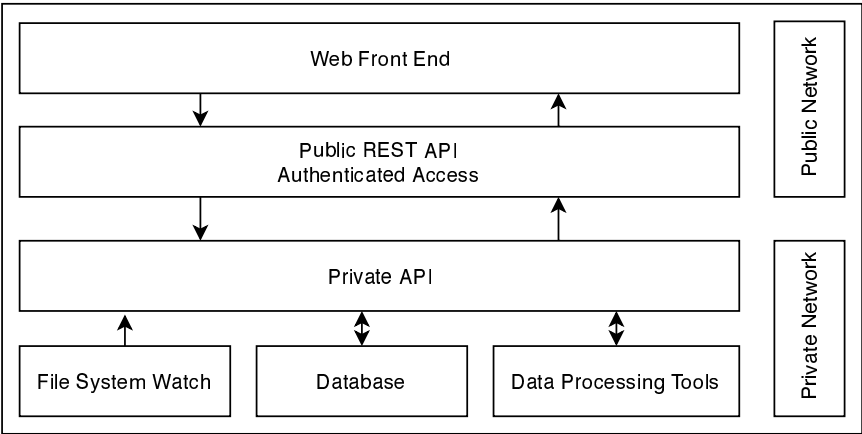


Figure 1. Simplified schematic representing the hierarchical structure and relation between components for the proposed design. Several components were intentionally omitted due to spacing.

3.1. Data Reduction Package

This is based on the offline GSP Python package. The modifications required for the real-time version are intended to enable asynchronous data processing as well as control, this could be achieved by adding a private REST API that would enable modifying the pipeline’s settings needed to operate under different observing strategies. It has to be relatively simple and able to work independently and automatically, allowing intervention by request from the Public API. The allowed control hooks in the private API should be minimal so that authentication is not required, though the access should be constrained to the Observatory’s private network. Some of the controls that need to be included are: turn on or off file system event watching routines, alter settings, report errors. It should also handle a light database for redundancy in case the web server fails. In terms of data processing most of the routines are already implemented in the offline version, but there are a couple of things still missing, such as optimal extraction Marsh (1989) and Horne (1986) flux calibration, deblending of multiple sources, low signal-to-noise extraction.

As is explained in Torres-Robledo et al. (2019) our philosophy is to rely as much as possible on Astropy's code Astropy Collaboration & Astropy Contributors (2013) and Astropy Collaboration & Astropy Contributors (2018) therefore, some of the code we had to develop ourselves because there was nothing equivalent implemented yet in Astropy. We plan to add these pieces of code as our contribution to the appropriate Astropy Package.

3.2. Public REST API

Publishing a web application is not a simple task, with security been the biggest point of concern, but also that the application be stable and reliable. Fortunately, there are plenty of tools that allow us to simplify these tasks, in fact, most of them, which is important because we don't have the resources to hire an entire team of developers experts on these very specific tools. The scope of the services provided by the public API should be end-user oriented only, such as, secure authentication and channeling communications with the private API.

3.3. Web Front End

A highly responsive website is favored over a local GUI for one simple reason: most of the GHTS users are working remotely, while local users will still benefit from it. Though more difficult to implement, there are several benefits that come with a web front end, for instance: adaptive layout, user experience less dependent on connection quality, less bandwidth usage and of course taking advantage of the interactive experience that web technology allows.

We have not yet decided what are the tools (stack) that we will use, but the criteria for selecting them are: being well documented and easily testable. By *easily* we mean that there have to be good tools available and a good testing philosophy behind its development. Testability is highlighted here but it applies to all the code of our project.

Acknowledgments. The authors would like to acknowledge the important contribution from Bruno Quint, David Sanmartim and Tina Armond. This research made use of Astropy, a community-developed core Python package for Astronomy Astropy Collaboration & Astropy Contributors (2013) and Astropy Collaboration & Astropy Contributors (2018) This work has been developed at the Southern Astrophysical Research (SOAR) telescope, which is a joint project of the Ministério da Ciência, Tecnologia, Inovação e Comunicações (MCTIC) do Brasil, the U.S. National Optical Astronomy Observatory (NOAO), the University of North Carolina at Chapel Hill (UNC), and Michigan State University (MSU).

References

- Astropy Collaboration, & Astropy Contributors 2013, A&A, 558, A33. 1307.6212
- 2018, AJ, 156, 123. 1801.02634
- Clemens, J. C., Crain, J. A., & Anderson, R. 2004, in Ground-based Instrumentation for Astronomy, edited by A. F. M. Moorwood, & M. Iye, vol. 5492 of SPIE, 331
- Horne, K. 1986, PASP, 98, 609
- Marsh, T. R. 1989, PASP, 101, 1032
- Massey, P., & Hanson, M. M. 2013, Astronomical Spectroscopy, 35
- Torres-Robledo, S., Briceno, C., Quint, B., & Sanmartim, D. 2019, in ADASS XXVII, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 533

Session IV

Management of Large Science Projects

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Hit the Ground Running: Data Management for JWST

Anastasia Alexov, Mark Kyprianou, and Catherine Kaleida

Space Telescope Science Institute, Baltimore, MD, USA; alexov@stsci.edu

Abstract. As the launch of James Webb Space Telescope (JWST) approaches, a team of engineers and scientists is hard at work developing the Data Management Subsystem (DMS) for JWST with its cadre of complex imaging and spectral instruments. The DMS will perform receipt of science and engineering telemetry data; will perform reformatting, quality checking, calibration, and data processing; will archive the data; will have tools for retrieving the data; will have the capacities for reprocessing the data; will have external/public calibration tools; will provide user notification, search, and access tools for JWST science and engineering data; will distribute data to the end user; provide extensive user analysis/visualization tools; and, will provide support for contributed data products from the community. We will give an overview of the software components, the hardware they run on, the programming languages/systems used, the complexity of the tested end-to-end science data flow, the current functionality of the system and what is to come for the JWST Data Management Subsystem in preparation for launch.

1. Data Management Subsystem (DMS) for JWST

JWST DMS is comprised of 39 FTE staff. The primary focus is development work on JWST end-to-end data processing, calibration, archiving, data access services and data analysis tools. This work is spread over 5 Engineering Branches with over 100 staff. The DMS is one of six subsystems within the Science and Operations Center (S&OC) located at the Space Telescope Science Institute (STScI). The other subsystems are: The Flight Operations Subsystem (FOS), Project Reference Database Subsystem (PRDS), Wave Front Sensing and Control (WFS&C) Software Subsystem (WSS), Proposal Planning Subsystem (PPS), and Operations Script Subsystem (OSS).

All roads from the S&OC subsystems meet at DMS, requiring massive coordination. There are 2.5 FTEs for Technical and Project Management to interface with over 120 staff across other JWST Subsystems and STScI Divisions and Branches. The DMS Leads spend over 1/2 the week in more than 20 meetings in order to coordinate with these people, across a wide variety of topics, across components, and across subsystem interfaces. In addition to delivering the system to process JWST data and serve it to the community, DMS is moving from a Waterfall to an Agile methodology using 2-week Sprint cycles. This is an enormous cultural change, which requires additional training and time to adapt for a large number of staff. In order to improve our efficiency and the ability to catch issues quickly, we are also utilizing Continuous Integration (CI).

DMS has 704 requirements to complete under its NASA contract, of which 97% are complete as of November 2018. DMS is using cryo vacuum test data to develop code to and to verify these requirements; however, these data do not cover “typical”

science use cases. DMS needs test data from the end-to-end simulator to complete the remaining requirements to the best of our ability prior to launch. This simulator is just coming online in fall 2018; issues are being ironed out in order for DMS to receive the remaining datasets it needs to complete coding and requirements.

Requirements are delivered as software functionality in large “Builds.” There are 14 DMS software components, or “groups,” with specific functionality. All these 14 development groups deliver a variety of software, which all combine into a large integrated software build. Builds are installed approximately twice per year; DMS has been delivering releases for over 6 years. Patches to Builds are installed more frequently in order to fix urgent issues. DMS is moving towards being able to install Builds on a monthly basis instead of having to wait 6 months for new functionality.

Each DMS build comes on a String/Environment containing nearly 20 servers, which has all 14 components installed. DMS maintains 4 sets of Strings, one for each of these main purposes: Development, Test, Shadow and Operations. The Shadow String is a copy of Operations – it can be used to perform and verify operational installations before making them “live” on the Operational String.

The major software technologies utilized by the software developments teams are: HTCondor - used as the pipeline infrastructure; Python - used as the primary programming language for the pipeline, calibration, and data analysis tools; Django & Apache - used for the pipeline workflow manager UI; numpy & Astropy – used and contributed towards the calibration pipeline; (Astro)Conda – packaging/distribution of the calibration and data analysis tools; MySQL & SQLite – used for small database needs; MS SQL Server – used for large database needs; C# & Javascript – used as the client server architecture for the Archive User Interface (AUI); Hibernate – used for database, distribution and the storage broker; Java & Scala – used for the archiving, distribution and operator tools UI; Confluence & JIRA – used for operational procedures, notes, shift reports, bug report ticket tracking, etc.

Additionally, the following major system technologies have been utilized by DMS: a 1 PB expandable Isilon is used for data processing and storage while SSDs are used for fast storage in databases; Single Sign-On is implemented at STScI using Shibboleth (Alexov et al. 2017), and Federation is being added in 2019; there is a mixture of IIS servers as well as Linux servers – all servers are Virtual Machines (VMs) and can be spun up in a 1/2 a day; DMS uses a service-based architecture; the Mikulski Archive for Space Telescopes (MAST) has a programmatic interface and is VO-compliant; all the 14 development groups use Github for software configuration management; and some of the automated testing tools used by DMS are Jenkins, Artifactory and Selenium.

2. Current Functionality

As of fall 2018, the system is able to perform the following: (1) receives and archives Science and Engineering data; (2) processes Science data through calibration and some combined products (Alexov et al. 2019) (3) provides search and distribution to end-users; (4) provides monitoring tools for operations (Kaleida et al. 2019); (5) provides tools for data re-calibration and post-processing (Bushouse et al. 2019), (Diaz et al. 2019); (6) provides Data Analysis tools (Ferguson 2019).

2.1. Archive User Interface, aka “The MAST Portal”

The MAST Portal (<https://mast.stsci.edu>) will be used to search and retrieve four major types of JWST data. (1) JWST Science Data as well as Science Instrument Tables can be searched by program, position, or using advanced filtering on instrument metadata parameter(s) to hone in on the dataset(s). There are several download options available: ZIP, cURL, Batch, etc.. (2) Guide Star/Jitter data can be found associated with its’ science program, as well as by ID if no science was taken. (3) JWST Engineering Data will be made public; users can search by mnemonic (parameter) for a date range. (4) WFS&C Optical Path Difference (OPD) data can be found programmatically by the STScI Telescopes team, used to align and focus the mirrors. AstroTurf is the MAST API <https://astroquery.readthedocs.io> (Brasseur et al. 2019) which can be used to access JWST data programmatically. MAST is VO-compliant. The MAST demos can be found on the “STScIMAST” YouTube channel: <https://www.youtube.com/user/STScIMAST>.

A new feature recently added for HST data in MAST, and will also be available for JWST data, is the Subscription and Notification Service (DuPrie et al. 2019). Archive users can sign up for subscriptions based on a proposals or observations or position-on-the-sky; once signed up, the users can be notified when these data become available.

2.2. Data Analysis Tools

Currently there are no flight JWST datasets. Nevertheless, the JWST Data Analysis team is moving forward in building a Python-based tool suite which can be used by the science community to learn the JWST tools using datasets from other observatories and give feedback to development in order to improve the software.

STScI’s DMS Data Analysis Tools offer the following: (1) Library infrastructure (i.e. Astropy, gWCS (Dencheva et al. 2019)); (2) Spectral analysis tools (MOSViz, SpecViz); (3) Photometry analysis tools (photutils); (4) Visualization tools (stginga (Lim et al. 2019), CubeViz, Glue); (5) Training and Documentation (conferences, workshops, Jupyter notebooks online).

3. Lessons Learned

The JWST DMS Leads would like to share the following lessons learned from managing DMS: (1) High fidelity, end-to-end simulators to create data are vital for development and to exercise the S&OC. (2) Automated testing is essential to catch problems: add tests within the code (Python), use tools such as Jenkins, Artifactory and Selenium, and use end-to-end regression testing with simulated data. (3) Effort spent up front in Interface Control Docs (ICDs) is well worth it. (4) Integrate your components/systems as early as possible and often! (5) Align your release process across components/systems so that they can be tested together and in-step. (6) Have a well-defined installation and patch process with designated sign-off; have a plan to revert in case of emergency. (7) Use Test or Shadow Strings to pre-test a release or flip between Shadow and OPS environments for operations. (8) Get the baseline (“vanilla”) calibration accomplished first! Fend off scope creep - calibration is never-ending and will be improved and tweaked endlessly to get “chocolate.” (9) Combined higher-level products (e.g. mosaics, dithers, background subtraction, extracted and fully calibrated spectra from multi-spec instruments and grisms) are really hard! Simplify your de-

sign when possible. (10) If using Agile/Scrum for large systems, establish a Scrum of Scrums or similar for interdependencies across teams. (11) Large systems are hard to manage; put in management structure, checkpoints and communication to help with the coordination effort. (12) More interfaces/subsystems multiplies the dependencies and complexities; requires more resources for testing and coordination. (13) Establish and document procedures and processes so teams know what is expected, but don't be overly restrictive with respect to team management.

4. Summary

JWST DMS is nearly ready to receive, process, calibrate, archive and distribute JWST data. As with any space mission, there will be many unknown unknowns which will need to be fixed quickly after launch. The JWST DMS is preparing for launch via test rehearsals using the end-to-end data simulator; these are staffed by JWST operations personnel who are supported by development in order to help operators learn to use the system and for development to be on the front lines when issues are found processing data through the system. With the additional two years of launch delay DMS development will be able to improve the system, performance, tools, infrastructure, and user interfaces as well as add additional functionality to the system.

Acknowledgments. I would like to thank the entire JWST DMS team comprised of over 100 engineers and scientists. I would especially like to thank Mark Kyprianou and Catherine Kaleida - we had a great time leading the JWST DMS project!

References

- Alexov, A., Swade, D., et al. 2019, in ADASS XXVII, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 677
- Alexov, A., et al. 2017, in ADASS XXV, edited by Nuria P. F. Lorente, Keith Shortridge, and Randall Wayth (San Francisco: ASP), vol. 512 of ASP Conf. Ser., 93
- Brasseur, C., et al. 2019, in ADASS XXVII, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 97
- Bushouse, H., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 543
- Dencheva, N., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 551
- Diaz, R., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 305
- DuPrie, K., et al. 2019, in ADASS XXVII, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 641
- Ferguson, H. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 269
- Kaleida, C., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 175
- Lim, P.-L., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 325

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Gaia DPAC Project Office: Coordinating the Production of the Largest Star Catalogue

Gracia-Abril, G.,^{1,2} Teyssier, D.,^{1,3} Portell, J.,^{1,4} Brown, A.G.A.,⁵
Vallenari, A.,⁶ Jansen, F.,⁷ and Lammers, U.⁸

¹*Gaia DPAC Project Office, ESAC, Villanueva de la Cañada, Madrid, Spain*

²*Astronomisches Rechen-Institut, Zentrum für Astronomie, University of Heidelberg, Heidelberg, Baden-Wurtemberg, Germany*

³*Telespazio Vega UK Ltd for ESA / ESAC, Villanueva de la Cañada, Madrid, Spain*

⁴*Dpto. Física Quàntica i Astrofísica, Institut de Ciències del Cosmos (ICCUB), Universitat de Barcelona (IEEC-UB), 08028 Barcelona, Spain*

⁵*Leiden Observatory, Leiden University, Leiden, The Netherlands*

⁶*INAF - Osservatorio astronomico di Padova, Padova, Italy*

⁷*Mission Operations Division, Operations Department, Directorate of Science, European Space Research and Technology Centre (ESTEC/ESA), Noordwijk, The Netherlands*

⁸*European Space Astronomy Centre (ESA/ESAC), Villanueva de la Cañada, Madrid, Spain*

Abstract. The European Space Agency mission Gaia is creating the most complete and accurate map of the Milky Way. It was launched in December 2013 to its orbit around the L2 Lagrange point. The extension of the mission until the end of 2020 has recently been approved. The Gaia archive is published and accessible to the astronomy community from its central hub at ESAC (ESA) (<http://gea.esac.esa.int/archive/>) and many other data centres around the world. Two Gaia Data Releases have been completed so far. The number of scientific papers based on Gaia Data Release 2 since it was published in April 25th, 2018, more than 400, gives an idea of the impact Gaia is having in the astronomy community.

The DPAC Project Office (PO), together with the DPAC Executive, is responsible for the coordination of the Data Processing and Analysis Consortium, DPAC. Created in 2009, the PO is a key point to maintain the consortium closely coordinated and focused on the common goal of creating the best possible Gaia archive. A well balanced composition of the PO, including management, scientific and engineering expertise, is fundamental to accomplish its role in a very complex structure as DPAC, where common industry management practices are not fully applicable and speaking a common language to the scientists and engineers is critical.

1. Gaia data reduction

Gaia is continuously scanning the sky taking, mainly, one dimensional measurements of all the stars crossing the field of view of its two telescopes. Gaia data, collected by its three instruments consisting of astrometry, spectrophotometry and medium resolution spectroscopy, are downloaded to Earth daily and processed immediately. The main goal of the daily processing is to assess the health of the payload to detect any issue affecting the quality of the scientific data and react quickly to minimize any data, or quality, loss. Besides the satellite monitoring, the daily pipelines also generate some initial calibrations to be used in the cyclic data reduction. Finally, two dedicated daily pipelines raise alerts on events which may require immediate follow up from ground, photometric science alerts (e.g. possible new supernovae) and detections of unknown asteroids. In both cases these alerts are immediately published, and made available to the community, on line ¹.

Data reduction after the daily processing is iterative. In each processing cycle the data produced in the previous cycle, and the new daily data collected until a given date, are consistently processed by all pipelines. This consistency is required because of the dependencies between the different subsystems: photometric and spectroscopic processing needs astrometric results, downstream systems need the data from astrometric, photometric and spectroscopic processing. Likewise, these three upstream pipelines need some results from the downstream systems which are only available to them in the following processing cycle.

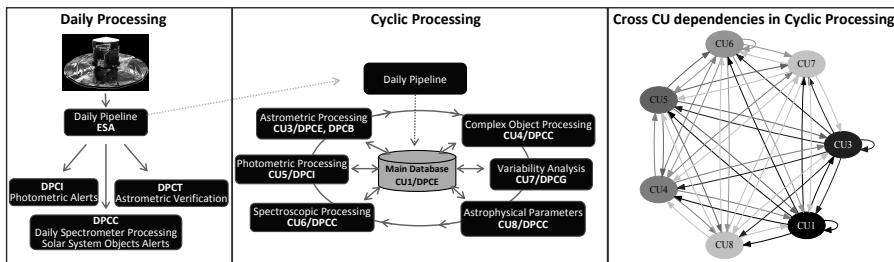


Figure 1. Schematic view of Gaia data processing and cross CU dependencies.

2. The Data Processing and Analysis Consortium, DPAC

The Data Processing and Analysis Consortium (DPAC) is in charge of the above described reduction of the Gaia data, from the raw telemetry to data products usable by the astronomers. It is also responsible, together with ESA, for the production of the Gaia Data Releases. DPAC is an international consortium with more than 400 members, distributed in 83 institutes in 22 countries plus the European Space Agency (ESA). The development of the processing software and the scientific validation of the data produced is the responsibility of nine *Coordination Units* (the CUs). Each CU is composed

¹Photometric Alerts <http://gsaweb.ast.cam.ac.uk/alerts/home>, Solar System Objects Alerts <https://gaiafunso.imcce.fr/index.php>

of several dozen members, spread around many (academic) institutes in various, mostly European, countries, and is in charge of a specific aspect of the data processing: Architecture and Common Tools (CU1), Simulations (CU2), Astrometry (CU3), Multiple systems, Extended and Solar System Objects (CU4), Photometry (CU5), Spectroscopy (CU7), Variability (CU7), Astrophysical Parameters (CU8) and Archive Preparation (CU9).

The actual data processing is done in the following six data processing centres (DPCs): ESAC (DPCE, Madrid), BSC (DPCB, Barcelona), CNES (DPCC, Toulouse), Institute of Astronomy (DPCI, Cambridge), ISDC (DPCG, Geneva) and ALTEC (DPCT, Turin). Each DPC receives and operates the software modules from one, or several, CUs. Each processing pipeline only runs in one of the DPCs. At the end of each processing cycle data products generated in every DPC are transferred to the Main Data Base (MDB) located at DPCE. All the data types are combined and redistributed back to all DPCs to serve as input for the next processing cycle.

3. The Gaia DPAC Project Office

The Gaia DPAC project office was created in 2009. Until the Gaia launch, the PO work was strongly linked with CU1, the coordination unit in charge of designing the general architecture of the consortium. The CU1 leader was also member of the PO. Among other tasks, some of the activities the PO did before launch are:

- Definition of the cross CU interfaces, and writing the DPAC Operations Interface Control Document.
- Coordination of development of the sub-systems which form the daily pipeline. These systems were critical as they had to be ready from the very first minute after launch to be able to process the data during commissioning.
- Preparation of operations rehearsals, tests aimed at simulating daily normal operations after launch in realistic conditions, and the end-to-end tests of the cyclic processing pipelines.
- Preparation, together with the CUs and DPCs, of the documentation required for the ESA reviews.
- Cross CU coordination of QA activities and DPAC level risk management.

During the nominal mission the PO activities have evolved to focus on four main types of activities:

- Coordination of DPAC operations: creation and maintenance of the operational schedule. The operational flow must take into account all the dependencies between the processing pipelines, including deadlines for deliveries between the different groups. Deliveries of preliminary validation data are also needed in the development and test phases. The top level DPAC priorities must always be taken into account when planning the individual DPCs operations.
- Technical support to the CUs and DPCs (together with CU1): provide technical support and expertise to the sub-groups and advise on the use of common tools. The PO also fosters technical synergies to avoid, when possible, duplication of work, proposing applying similar solutions to equivalent problems.

- Scientific interfaces: The dependencies between processing pipelines, with all systems using data produced by upstream pipelines, require a good understanding of data caveats and clear flagging of outliers and special cases. DPAC runs in operational mode, processing the Gaia data, but is also in continuous development to improve the results and reduce the systematic errors. Data producers are well aware of the remaining weaknesses in their software but have limited knowledge of the impact of these caveats in the downstream systems. Although the technical interfaces may be perfectly defined, identifying and communicating data issues and caveats between the units is critical.
- Data release preparation: the publication of the data in the Gaia Data Releases requires additional communications channels. The data in the MDB is optimised for internal DPAC use. However the archive data model has to serve a different purpose, which is usability by the end users of the archive. Transferring the data from the MDB to the archive requires data conversions and data filtering. Both conversions and filters must be communicated clearly to CU9 by the rest of DPAC. Thanks to its overall knowledge of the operations and of the data produced, the PO has played a very active role in the definition of such interfaces.

The composition of the PO has evolved since it was created. A well balanced mixture of technical, management and scientific expertise is needed. All members of the PO have at least experience in two of these aspects, either scientists with coordination or management experience, or engineers with experience in science. These mixed profiles are essential to communicate efficiently with all the subgroups. The PO role can only be achieved if its members are able to understand the problems and speak the language of the scientists and engineers.

Communication is a key word in a large consortium like DPAC. Miscommunication and lack of common understanding of problems are extremely expensive in time and resources. This was a real problem at the start of Gaia Operations. The PO participates in subgroups meetings and has regular teleconferences with the leaders of each processing pipeline and of the DPC where those run. This direct communication channel has demonstrated to be very useful. The PO has direct visibility of the progress, difficulties and open issues in all subgroups. In some cases the PO can provide direct support to them. In other cases, the PO opens new communication channels with other groups who may have had, and solved, similar problems. Finally, thanks to these frequent exchanges, the PO identifies schedule or data issues which may have impact in downstream systems and shares them with the concerned teams. This early communication helps the development teams to plan their activities and pro-actively prepare their systems to handle originally unexpected issues before they receive the data.

The PO does not have real executive power. Besides the horizontal communication, described in the previous paragraph, the PO also acts as a vertical communication channel, ensuring discussions are held at the right level. The PO regularly informs the DPAC Executive (the group of CU leaders) of the overall progress, relevant issues, risks and proposes solutions so the executive decisions can be taken as appropriate.

The Gaia DPAC PO is essential in planning and coordinating DPAC activities. It has brought significant added value to the consortium, reducing the non technical work load of scientists and optimizing the resources by diminishing the effort wasted due to lack of coordination or miscommunication. From our experience in Gaia, a similar structure is mandatory in any large scientific consortia as DPAC.

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

The VLA Sky Survey – Operations, Data Processing and Archiving

Mark Lacy,¹ Claire Chandler,² Amy Kimball,² Steve Myers,²
Kristina Nyland,¹ Stephan Witz,² and the VLASS Team

¹*National Radio Astronomy Observatory, Charlottesville, VA, USA;*
mlacy@nrao.edu

²*National Radio Astronomy Observatory, Socorro, NM, USA*

Abstract. The VLA Sky Survey (VLASS) is an ambitious project to image the entire sky visible to the VLA in three epochs. The Survey is being carried out at a frequency of 2-4GHz in full linear polarization (Stokes I,Q,U) at a resolution of 2.5 arcseconds, making it the highest angular resolution all-sky radio survey ever attempted. VLASS will collect 0.5 PB of raw data, and 0.3 PB of Basic Data Products (calibration tables, images, and catalogs) will be created and archived at NRAO. There are also opportunities for external groups to create higher level Extended Data Products. In this paper, we present a summary of the scheduling, operations, data processing, and archiving of the survey.

1. Introduction

The VLA Sky Survey is motivated by the need to provide a modern survey of the radio sky at comparable resolution to all-sky surveys in the optical (e.g., PanSTARRS and LSST) and infrared (e.g., WISE), including time domain, spectral, and polarization information. This has been made possible by advances in technology and observing techniques since the original NRAO Sky Survey (NVSS) (Condon et al. 1998) and the Faint Images of the Radio Sky at Twenty Centimeters (FIRST) survey (Becker et al. 1995). The advent of wide band backends allows a full octave of data to be taken at gigahertz frequencies, and on-the-fly interferometry (Mooley et al. 2018) allows the rapid imaging of large areas of sky without the overheads associated with settling the antennas at each pointing.

2. Survey observations

The on-the-fly mode used for VLASS observations is deployed in $\approx 40 \text{ deg}^2$ tiles. In a typical observation, the antennas are scanned in a raster pattern with rows of length $\approx 10 \text{ deg}$. along the Right Ascension direction, separated in Declination by half the primary beam width (7.2 arcmin). Data are sampled every 0.45s, with the phase center of the array being stepped every 0.9s (corresponding to about a tenth of the primary beam width at the scanning rate of 3.3 arcmin/s). Tiles are typically paired to make scheduling blocks of 4hr duration, sufficient to obtain a good polarization calibration from a calibrator moving through transit, while also being short enough that they can

be easily slotted into the dynamic scheduling of the VLA. Observations and processing are controlled by the Survey Manager, a piece of software that includes a database of observations and their processing state.

3. Data processing

Each epoch of VLASS will generate 200 TB of raw visibility data, for a total of 600 TB. These data will be processed by three pipelines. The first, "Quicklook" pipeline performs a calibration of the data and makes an image in Stokes I only, relatively coarsely sampled (1 arcsec per pixel). The main aim of these data products is to allow rapid identification of transients, with a goal of one week between observations of a tile and the completion of its images. The imaging therefore uses some approximations that currently limit the flux density accuracy of sources to $\approx 15\%$ and the positional accuracy to ≈ 0.3 arcsec (though it is possible future versions of the Quicklook pipeline may improve on this). The Quicklook images are thus most useful for quality assessment of the observations, transient detection, and morphological studies. The second, "Single Epoch" pipeline performs more accurate, better sampled imaging (0.6 arcsec per pixel), and, in addition to Stokes I images, will also make coarsely sampled (128 MHz channel width) cubes in Stokes I, Q, and U (though we may only be able to afford to store cutouts around bright sources). The Single Epoch pipeline also makes spectral index images, and provides catalogs of source components. The final, "Cumulative Imaging" pipeline is designed to combine the data from different epochs to produce deeper images. This pipeline will also make high resolution (2 MHz channel width) cubes around the brightest objects in the survey, and include spectral curvature information. All calibrations, images and catalogs are quality assured by data analysts before being placed into the archive.

4. Archiving

The raw data and the outputs from the Quicklook, Single Epoch, and Cumulative pipelines will be archived at NRAO. A new NRAO Archive Interface is being developed to serve both VLASS data and data from other NRAO/AUI facilities such as the VLBA and ALMA, and the Greenbank Observatory. This interface will allow searches on all types of data, and also a limited number of simple operations on raw data to make it more amenable to user processing, such as transformation into a CASA measurement set, application of calibration tables and flags derived from the pipelines, and smoothing in time and frequency.

5. Enhanced data products

Early in the planning of VLASS it was recognized that NRAO alone does not have the resources to make and store all the possible products from VLASS. The data processing scheme described above will result in approximately 600 TB of products to accompany the 600 TB of raw data, about 60% of the size of the current VLA archive. Unless the cost of storage decreases significantly in the next few years, any additional products must be made and stored elsewhere. The Canadian Initiative for Radio Astronomy Data Analysis (CIRADA) and the South African Inter-University Institute for

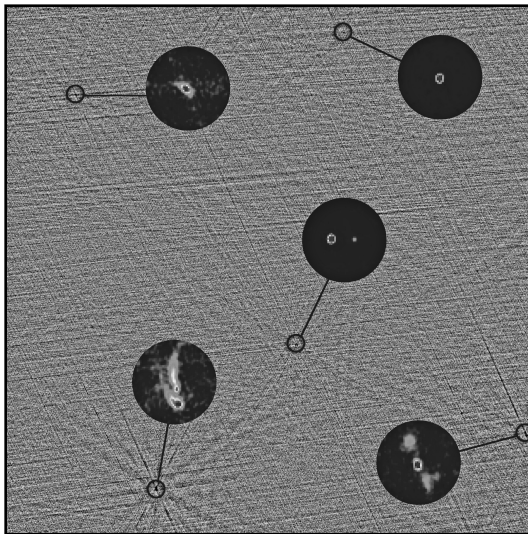


Figure 1. Example VLASS Quicklook image, with magnified views of bright sources. Each Quicklook image is 1 deg^2 .

Data Intensive Astronomy (IDIA) both plan to produce and store enhanced products made from VLASS data.

6. Commensal Experiments

While the main survey is being carried out on the VLA, two other projects will be operating simultaneously. The *realFAST* project (Law et al. 2018) is designed to identify candidate Fast Radio Bursts (FRBs) using real-time processing. The VLITE Commensal Sky Survey (VCSS) uses the VLITE low frequency system at prime focus (Clarke et al. 2016) to image the sky between 320 and 384 MHz. The resolution of VCSS is about 20 arcsec and the sensitivity 3 mJy/beam. Products from both projects will be made available from the NRAO archive.

7. Early Observations

Observations of a pilot survey were conducted during the summer of 2016. The pilot survey was designed to test the VLASS observing strategy, demonstrating successfully that the data could be acquired as planned, and identified a number of issues with the observations that were addressed before the survey proper began.

The first half of the sky was observed in the first epoch (VLASS 1.1) from 2017 September until 2018 February. Calibration mostly followed standard procedures in ra-

dio interferometry. To address issues with gain compression arising from severe radio frequency interference (RFI), though, a new step was added to the calibration procedure. This step uses the data from a switched power source that adds a known power to the data for half the time at 10 Hz and allows an estimate of any variation in the system gain to be tracked.

The Quicklook imaging pipeline has now been run on all the VLASS 1.1 data, and the images are accessible from the NRAO archive (e.g., Figure 1). The next step involves the finalization of the Single Epoch pipeline, and the taking of the second half of the first epoch of data (scheduled to start in 2019 February).

References

- Becker, R. H., White, R. L., & Helfand, D. J. 1995, *ApJ*, 450, 559
- Clarke, T. E., Kassim, N. E., Briske, W., Helmboldt, J., Peters, W., Ray, P. S., Polisensky, E., & Giacintucci, S. 2016, in *Ground-based and Airborne Telescopes VI*, vol. 9906, 99065B
- Condon, J. J., Cotton, W. D., Greisen, E. W., Yin, Q. F., Perley, R. A., Taylor, G. B., & Broderick, J. J. 1998, *AJ*, 115, 1693
- Law, C. J., Bower, G. C., Burke-Spolaor, S., Butler, B. J., Demorest, P., Halle, A., Khudikyan, S., Lazio, T. J. W., Pokorny, M., Robnett, J., & Rupen, M. P. 2018, *ApJS*, 236, 8. 1802.03084
- Mooley, K. P., Frail, D. A., Myers, S. T., Kulkarni, S. R., Hotokezaka, K., Singer, L. P., Horesh, A., Kasliwal, M. M., Cenko, S. B., & Hallinan, G. 2018, *ApJ*, 857, 143. 1803.07092



Not the first time at ADASS, but well worth a repeat. Around the table from left to right are Tom McGlynn, Bruce Berriman and Igor Chilingarian. Jim Lewis in cameo. (Photo: Peter Teuben)

Session V

Science Platforms: Tools for Data Discovery and Analysis from Different Angles

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Hubble in the Cloud: A Prototype of a Science Platform at STScI

Ivelina Momcheva and Arfon Smith

Space Telescope Science Institute, Baltimore, MD, USA; imomcheva@stsci.edu

Abstract. The availability of high-quality, highly-usable data processing and analysis tools is of critical importance to all astronomers, as is easy access to data from our archives. In this talk I will describe the approach to developing the prototype of a new cloud-based data management environment for astronomical data reduction and analysis at STScI. In this paper we discuss the use cases and requirements set by operational and scientific needs, examine the decisions we made, and demonstrate the prototype built at STScI.

1. Introduction

Data processing is a core task of many astronomical data centers and support institutions as well as a common task for the vast majority of scientists and their students. At STScI, we carry out data processing as part of our routine daily operations and provide the building blocks for our users to carry out processing and analysis themselves.

First and foremost, within data management it is our responsibility to process, store and distribute the data of our missions to the astronomical community. The Hubble Space Telescope (*HST*) is our largest current mission. *HST* carries out daily observations for calibration and scientific purposes and the data are continuously processed and archived. Furthermore, archived datasets are continuously kept up to date with improvements in pipeline algorithms and reference files. The same is true for other missions we support and will be true for the James Webb Space Telescope (*JWST*). The internal framework which executes these steps is complex and composed of many collaborating teams and interrelated systems developed over the course of two decades. The effort dedicated to data management is significant and any efficiencies and improvements would have far-reaching effects on our operations.

Our second, and equally important task is to support the astronomical community in carrying out scientific research. The paradigm we have converged on is to give users all the major puzzle pieces of data management: raw and processed data, reference and calibration images, and pipelines and data analysis tools. We provide the users with download and installation instructions as well as handbooks, cookbooks and tutorials on processing and analyzing data. We endeavor to support our community as much as possible through program scientists and an active help desk which allows us to understand what is difficult and where the pain points are. We see that the effort needed to put the pieces together by the end users is high and this means a high barrier to entry for new users and small teams which likely translates to missed science.

Despite this complexity, on the face of it the paradigm is working. Two measures that we use to determine success are the volume of data downloads and the number of

papers written. Figure 1 (left) shows the stored data volume versus the distributed data volume for several datasets in the Mikulski Archive for Space Telescopes (MAST). We see that for most of them, the volume of downloads far exceeds the data volume, i.e., datasets are downloaded many times over indicating high interest in the data we serve. We also show the number of published peer-reviewed papers per year based on *HST* data which has continued to grow since launch (Figure 1, right).

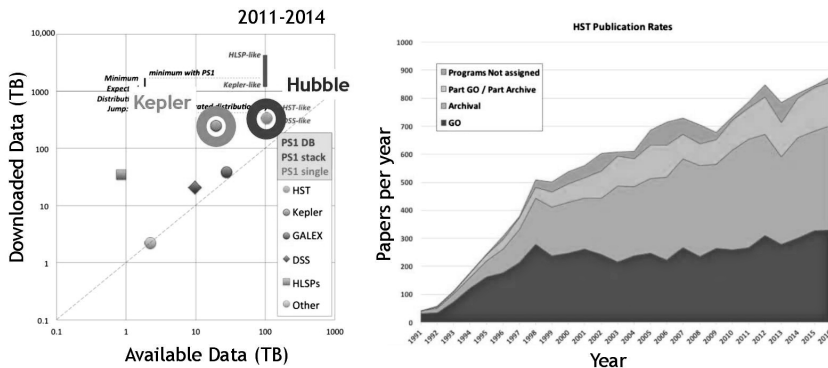


Figure 1. *Left:* Stored data volume vs. distributed data volume for MAST datasets in TB for the period 2011-2014. We highlight the *HST* and *Kepler* datasets. For these two datasets the download volume is ~ 4 and ~ 20 times larger than the stored data volume, respectively, demonstrating the great level of interest in these data. *Right:* Number of papers per year published with *HST* data. The areas shaded in blue, cyan, light green and dark green (bottom to top) show general observer (GO), archival, joint GO-archival papers and unassigned papers, respectively. Archival papers rival GO papers in volume and represent almost half of all publications based on *HST* data.

However, a combination of factors – the migration of *HST* away from IRAF, the imminent launch of *JWST* and challenging requirements for a novel *WFIRST* data processing environment – require us to innovate beyond the status quo and develop new solutions for supporting our communities. As we transition our data management today to *JWST*, and in the future to *WFIRST*, we are compelled to consider how the differences in the mission design will impact data processing and how we need to adjust accordingly. For example, *JWST* is unlikely to have the longevity of *Hubble* so how do we lower the barrier to entry for users to maximize science? *WFIRST* is different with observations focused on a handful of programs and science investigation teams expected to develop parts of the data processing pipelines. For *WFIRST* the pressing questions are: How do we maximize the archival science? How do we collaborate on pipelines with science teams? How do we enable the processing of large volumes of simulated data? And more broadly, how is science limited by the capabilities of data management and data analysis tools and how can we change that? How do we improve the provenance repeatability and reproducibility of data reduction and analysis? All of these questions can be directed both toward our internal operations and towards the tools and services we provide for users.

Science platforms are the answer to many of these questions as they allow us to unify the user experience, internal and external as well as across different research areas. Science platforms also have the ability to increase reproducibility and efficiency by using reusable portable components and doing less bespoke development.

While a number of different science platforms are in development, this paper focuses on the work done at STScI, our use cases and technology choices.

2. What is a Science Platform?

Science platforms are integrated, remotely-hosted environments which combine data storage, computational capabilities, software tools and interfaces which allows scientists to interact with the underlying component in order to access, visualize, subset and perform analysis on scientific datasets. A conceptual schematic of a science platform is shown in Figure 2 (see also LSE-319: LSST Science Platform Vision Document¹ Jurić, Ciardi and Dubois-Felsmann, 2017).

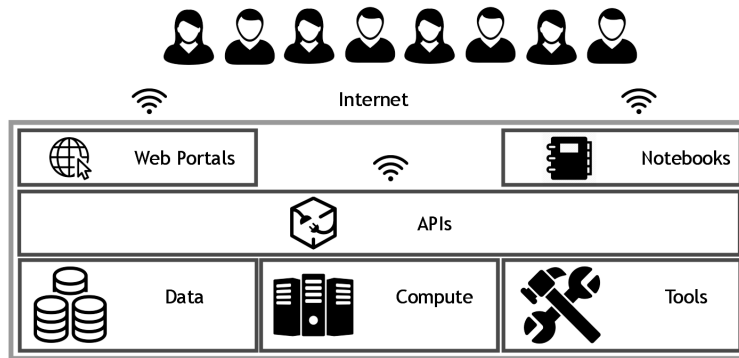


Figure 2. A conceptual schematic of the primary components of a science platform. This is a generalized version of the view presented in Figure 1 of LSE-319: LSST Science Platform Vision Document, Jurić, Ciardi and Dubois-Felsmann, 2017.

The key components of a science platform are:

- **Data:** High value datasets that cannot be reasonably downloaded by individual users.
- **Compute:** Computational capabilities that are flexible to support a wide range of scientific inquiries and data processing activities. Ideally, a user can specify on demand the resources they need.
- **Tools:** A predefined environment which contains a set of common tools to process, visualize and analyze the data.

¹<https://github.com/lsst-dmsst/LSE-319>

- **APIs, web portals, and notebooks:** Access to this infrastructure is through web portals and notebook-like interfaces which are built upon application programming interfaces (APIs) integrated with astronomical data services. Access to the underlying data is available directly through the APIs to enable a broad range of applications not served by web portals or notebooks. The platform is accessible through these multiple points to all users with an internet connection.

In the current paradigm of data processing, institutions such as STScI host these infrastructures but that is not necessarily the future. With appropriate investment in designing for portability, science platforms can be portable and can be deployed at multiple physical locations – on premise at the data centers, in the commercial cloud, at high performance computing centers (HPCCs) or on a local server. In fact, the same science platform can be instantiated at multiple physical locations to allow for different use cases – development versus running engineering data versus running simulated data. As a result of the modularity of science platforms, components can be customized and/or reused as needed.

3. Technological Convergence

Science platforms are not just a pie-in-the sky concept. They are in fact possible today, with existing technologies, and many commercial versions of them already exist, made possible by a convergence in development of four key pieces of technology over the last few years.

3.1. Notebook-driven Analysis

Notebook driven analysis is available across many platforms. Notebook-driven analysis is at the core of many major platforms such as RCloud, Apache Zeppelin, Google Colaboratory. Furthermore, user-generated notebooks are growing – as of late-2018 GitHub hosts over 3.5 million Jupyter notebooks², up from less than 100,000 in early 2016 indicating a rapidly growing base of users, including scientists, who are familiar with this type of analysis. As a result, tools in the notebook ecosystem are undergoing active development.

3.2. Compute is Commodified by Cloud Providers

Computing services are commodified by cloud compute providers. Commercial cloud providers allow for compute to be provisioned on demand, including access to the latest GPUs, and scaled up and down without a large upfront expense. This is a dramatic departure from the current paradigm where users need to procure physical compute resources which requires a substantial up-front infrastructure investments and locks them into a hardware solution. Cloud compute is in fact very well matched to the spiky demand of most astronomical projects. The use of commercial cloud providers is not to the exclusion of HPCCs. Science platforms can (and should) be hosted at HPCCs to enable the analysis of simulated datasets but a wider range of use cases is served by the flexibility of commercial clouds. However, additional work may be needed to enable the science platforms to run on HPCC infrastructure.

²<https://github.com/parente/nbestimate>

3.3. Software-defined Infrastructure Technologies are Mature(ing)

A key innovation in cloud computing is making infrastructure portable and software-defined infrastructure technologies are now reaching maturity. The combination of containerization software solutions such as Docker with orchestration technologies such as Kubernetes, an open-source system for automating deployment, scaling, and management of containerized application, means that infrastructure **can** be made portable and the same stack can be consistently deployed to many physical locations. Versioned, machine readable deployment templates ensure this consistency. Container orchestration allows for self-healing, scalable architecture that responds to user demand and only utilizes the needed resources.

3.4. A Rich System of Open Source Scientific Computing Tools

A rich open eco-system of scientific computing tools in Python has developed over the last decade. Core libraries such as `numpy`, `scipy`, `cython` and `dask` are at the heart of this eco-system and enable libraries that are focused on statistical techniques (`PyMC`), machine learning (`scikit-learn`) and image analysis (`scikit-image`) among others. Libraries with core functionality in specific scientific domains have also matured (`astropy`, `SunPy`, `biopython`). And finally, visualization libraries (`matplotlib`, `bokeh`) are allowing for true end-to-end scientific analysis in Python and also in the notebooks discussed above.

These commodity technologies and the rich eco-system means that there is a lot of off-the-shelf technology that we can use.

4. The STScI Science Platform

Starting in late 2017, an engineering team at STScI embarked on an exploratory project to build a prototype science platform that interfaces with our archive holdings. Here we describe the envisioned use cases and the technological decisions that were made based on those.

Science use cases for an STScI science platform:

- *Hubble* General Observer (GO) programs and archival (AR) programs span a broad range of data volume sizes, computational needs and scientific focus. Access to a science platform can lower the barrier to entry for researchers working on all programs by reducing the complexity of data reduction and analysis. Contrary to the typical thinking that science platforms are only for big data applications, the lower barrier to entry may impact small and medium programs the most since they may have fewer resources to tackle complex data management. This is particularly important for a limited lifetime mission like *JWST* where fast on-ramps for users will deliver more scientific productivity. A science platform should enable a broad range of science.
- New types of research techniques involving machine learning and artificial intelligence are entering astronomy. Some of these techniques are data- and/or compute-expensive. A science platform should be able to support the use of these techniques and allow access to different types of compute (e.g., GPUs).

- Analysis of simulated observations is key to survey-type missions such as *WFIRST* which expect the simulations volume to exceed the observations volume by orders of magnitude. These data will be generated at HPC and science platforms need the portability to run next to them.
- Joint pixel-level data analysis between large area surveys is a commonly discussed use case for science platforms. Joint processing will be greatly simplified if we have the ability to encapsulate the software stacks of different missions and to install them at the location of the data.

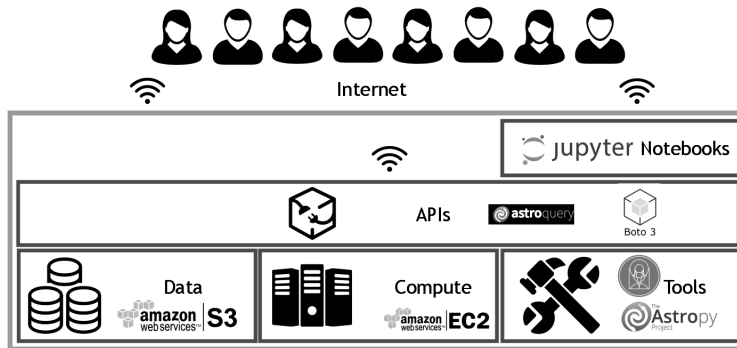


Figure 3. A schematic view of the STScI science platform described here, indicating the primary technological solutions chosen for each element.

Operational use cases for STScI science platform:

- While all of our data management infrastructure is currently on-premise, future infrastructure solutions may involve commercial cloud or hybrid cloud. Such solutions provide more flexibility to spiky demand driven by, for example, reprocessing campaigns.
- Collaboration between different teams developing and testing parts of the data management environment is difficult in the current model. For example, instrument teams and external scientists do not have access to the same environment that pipeline software developers do. Diaz & Marin (2019) discuss this issue in their contribution. A science platform should provide a common shared environment to such collaborating groups.
- Data processing centers frequently run large scale processing campaigns to update data and to create new data products. Examples of such campaigns are upgrading astrometric solutions, creating large-scale mosaics and deriving archive scale catalogs such as the Hubble Source Catalog (HSC). An example of creating source catalogs on images from the *Hubble* archive was shown by Momcheva (2019) using AWS Lambda functions. Science platforms need to support the development and execution of such tasks.

- Machine learning can also be part of operations: machine learning can be used for anomaly detection or for improved processing based the image content.
- Internal operations handle public and proprietary/exclusive-access data. Permissions and user accounts need to be managed in such a way as to preserve that data access rights. Limiting science platforms to public data only will curtail their utility.

Based on these use cases, the requirements for the STScI platform are:

- Allows for flexible compute.
- Deployment is infrastructure-agnostic, can be built and torn down easily.
- Software stack is containerized. Multiple software stacks can be available.
- User authorization and permissions management need to be secure.

A final major requirement we set is that as much as possible all technology we use should be “off-the-shelf” and open source.

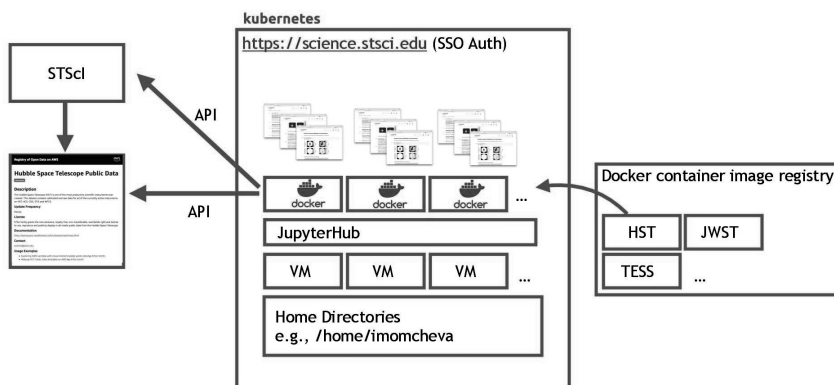


Figure 4. A schematic view of the internal operations and structure of the STScI science platform.

5. How to Build a Science Platform

A schematic of the current version of the STScI science platform prototype is shown in Figure 3 indicating the specific technology choices made for the different components. The platform is operated in Amazon Web Services (AWS) and takes advantage of the infrastructure and services provided there. All major cloud providers have feature parity with the services we use and these solutions are not unique to AWS.

Figure 4 shows a more detailed view of the internal operations and structure of the STScI science platform prototype. At the core of the platform is a JupyterHub instance. JupyterHub is a multi-user Jupyter server. The hub manages log-in, spawns new servers on demand and provisions the containers. User home directories are persistent

but single-user servers are shut down after a period of inactivity. The computational environment is managed by Kubernetes, an open-source container orchestration system for automating application deployment, scaling, and management. Authorization can be done either through OAuth (e.g., GitHub) or STScI Single Sign-On.

The **data** component of the science platform is a copy of ~120 TB of *HST* public data from the currently active instruments (ACS, COS, STIS, WFC3 and FGS). This dataset was staged in AWS S3 as part of the AWS Public Datasets Program.³ The AWS dataset is synced with the STScI archive as part of our operations and files uploaded (or updated) on AWS within 10-20 minutes of processing.

The **computational environment** is also hosted on AWS to provide scalability and allow for a range of computational resources which can be procured on-demand. The prototype has a single machine type but JupyterHub supports flexible machine types.

The **software stack(s)** available to users are based on pre-defined Docker images. A Docker image is a read-only template used to build containers. The images are stored in a Docker registry. Our goal is to have a range of containers for different purposes such as “*HST* Calibration Pipeline” containers, “*JWST* Calibration Pipeline,” “TESS data analysis tools.” Container images are compose-able and therefore make it easy for users to start with a base container and add to the stack (e.g., joint processing). In response to the use cases above we are working on creating on-demand containers from GitHub commit hashes which will allow for inter-team collaboration and rapid testing of software changes.

Access between the compute and the data is facilitated by an API. The Python client module `astroquery.mast`⁴ is a wrapper around the MAST API and the main search classes have methods to get paths and download data directly from AWS S3 to take advantage of the faster data transfer within an AWS data center.

Within JupyterHub users have access to Jupyter notebooks and the environment can be pre-populated with notebooks with example workflows. In order to create this content, we have been working with scientists across STScI to capture common data analysis tasks in notebooks.⁵ Style guides ensure a consistent look and feel across all content⁶ and, finally, continuous integration (CI) tests that notebooks are executable end-to-end and renders them as HTML.

As envisioned, all parts of the platform so far are off-the-shelf (except for the `astroquery.mast` module). We have been able to leverage open source and even contribute back upstream to projects.

A detailed description of deploying JupyterHub can be found in the popular “Zero to JupyterHub with Kubernetes” guide⁷ which we followed for our initial work. In order to simplify this workflow, we have encapsulated it in an Ansible playbook – Zero to Jupyterhub for AWS in Ansible⁸ – and released it to the community. By following this guide, building a new JupyterHub cluster can be done with a single command and

³<https://registry.opendata.aws/hst/>

⁴<https://astroquery.readthedocs.io/en/latest/mast/mast.html>

⁵<https://github.com/spacetelescope/notebooks>

⁶<https://github.com/spacetelescope/style-guides>

⁷<https://z2jh.jupyter.org/en/stable/>

⁸<https://github.com/spacetelescope/z2jh-aws-ansible>

takes about five minutes. At the end of this playbook you will have a basic JupyterHub installation suitable for use within a research group or collaboration.

6. Challenges and Future Directions

The work outlined above is still in its early stages and there remain a number of challenges to be solved.

A common expectation from users is that they can simply move their current workflows to a new environment. However, “lift and shift” workflows are not always possible and they do not take advantage of the new features and capabilities of the system. This is especially true for science platforms deployed on commercial cloud where “lift and shift” will not take advantage of the services developed by cloud providers. We need to educate the community of these new capabilities, provide exhaustive documentation and develop libraries of Jupyter notebooks showcasing different use cases in order to ease the transition.

Centrally managed resources always raise the question of privacy versus security. We need to monitor the system to ensure that there is no misuse of resources. The current STScI platform is only accessible to a handful of users but user management will become an increasingly important task as the number of users grows. We are slowly rolling the system out to a larger number of users and testing what their needs are. We believe it is important to avoid artificial quotas that can limit the usefulness of the platform. A related question is who pays for all of this: In some cases, missions and institutions can cover the expenses especially when related to internal operations. But in other cases the flexibility and ease of deployment may allow collaborations and research groups to host their own instances in which case grants can cover the costs. Cloud computing services are already an allowed expense for *HST* grants.

STScI has a wide range of use cases and a one-size-fits-all platform will not be suitable for all. Flexibility in machine types and containers will be key to meeting the needs of a broad range of users. A library of Docker containers needs to be created to meet those needs.

Collaboration is a key feature of software development and scientific research. Currently JupyterHub does not allow for direct collaboration on notebooks (such as in Google Colaboratory) or shared home directories. These features are in development. For these and other features we should consider how the needs of the astronomical community can drive technological development and how we can contribute to the open source tools we use for science platforms.

Many other teams are building similar platforms around similar use cases including SciServer⁹ (JHU), DataLab¹⁰ (NOAO, Fitzpatrick 2019), LSST, WholeTale¹¹ among others. There has been general convergence of technical solutions between projects and increased inter-project discussions including a workshop hosted at STScI in early 2018.¹² Through such collaborations we can exchange knowledge, share ap-

⁹<http://www.sciserver.org/>

¹⁰<https://datalab.noao.edu/>

¹¹<https://wholetale.org/>

¹²<http://www.stsci.edu/institute/conference/science-platforms>

proaches and, ultimately, ensure an uniform user experience across platforms and interoperability between components.

Another important point is that, despite their utility, not every astronomical data center needs to build, host or maintain a science platform. Significant gains in interoperability can be made by opening and supporting programatic access to archives via APIs, common protocols and client libraries (e.g., IVOA, *astroquery*).

7. Summary

Science Platforms allow users to run analysis next to data, real or simulated. While they are primarily discussed in the context of solutions to big data in astronomy in the future, they are also be key for many science and operations applications *now*. Small and medium data users will also benefit science platforms because they lower the barrier to entry by removing the need for users to install software, providing access to a range of computational resources and including tutorial notebooks to ramp up users. In the context of internal operations specifically, science platforms can improve the development cycle and allow us to expand capabilities beyond on premise resources. Across all applications, science platforms allow for reproducibility of data processing and analysis which will benefit both current and future missions.

As demonstrated by the science platform prototype developed at STScI, the technologies that underlie science platforms are maturing and open source solutions are available off-the-shelf. In summary, science platforms are possible today with existing off-the-shelf technologies and there are many ways both internal operations and the science community can benefit from their development now.

Acknowledgments. This work was supported by an internal STScI program: “Exploring next-generation data management environments” (PI: A. M. Smith) and is carried out in collaboration with J. Peek, M. Fox, J. Matuskey, C. Mesh, E. Tollerud, S. Crawford and others in the STScI Data Management Division.

References

- Diaz, R., & Marin, M. G. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 305
- Fitzpatrick, M. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 233
- Momcheva, I. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 671

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

The NOAO Data Lab: Design, Capabilities, and Community Development

Michael Fitzpatrick,¹ Knut Olsen,¹ Glenn Eyhaner,¹ Leah Fulmer,^{1,2}
Lijuan Huang,¹ Stephanie Juneau,¹ David Nidever,¹ Robert Nikutta,¹ and
Adam Scott¹

¹*NOAO, Tucson, AZ, USA; datalab@noao.edu*

²*University of Washington, Seattle, WA, USA*

Abstract. We describe the NOAO Data Lab, a new science platform to efficiently utilize catalog, image and spectral data from large surveys in the era of LSST. Data Lab provides access (through multiple interfaces) to many current NOAO, public survey and external datasets that combines traditional telescope image/spectral data with external archives, shares results and workflows with collaborators, allows experimentation with analysis toolkits and lets users publish science-ready results for community use. The architecture, science use-case approach to designing the system, its current capabilities and plans for community-based development of analysis tools and services are presented. Lessons learned in building and operating a science platform, challenges to interoperability with emerging platforms, and scalability issues for Big Data science are also discussed.

1. Introduction

The Data Lab was publicly released in 2017 to enable efficient use of image, spectral and catalog data generated by NOAO instruments and surveys. It does this by providing storage and compute resources close to the data, accessible from web, programmatic or commandline interfaces. Users can collaborate by sharing data and analysis workflows, new results can be published for community use via core Data Lab services. Hosted datasets include mirrors of standard reference catalogs (e.g. Gaia DR2, SDSS, AllWISE) for computational efficiency, survey results from NOAO instruments (e.g. Dark Energy Survey, DECam Legacy Survey) as well as internally-generated data such as the NOAO Source Catalog and numerous pre-computed crossmatch tables. Further, the NOAO Science Archive provides access to the raw and reduced frames used in generating detection images and catalogs are augmented with standard spatial-index or color values to provide a more useful and comprehensive dataset for analysis. The interfaces are designed for a range of user experience levels, providing an astronomer-friendly environment with functionality based on anticipated science use-cases. Data Lab is intended to serve as an incubator for experimenting with science workflows and to prepare the community for the era of LSST operations.

2. System Design Considerations

The total volume of data available is daunting (>2PB of images, tens of TB of catalogs), even subsets of the data would have been impractical to download by multiple users, so simply improving the archive facilities would not suffice. Designing Data Lab thus began with the question: “How will users want to access and interact with the data?” To answer this question, a range of use-cases covering solar system, galactic and extra-galactic science were used to derive functional requirements for the system. Interface requirements were derived by considering how these scenarios might be executed from the user’s desktop, a web browser or in the data center itself.

The system architecture fell naturally into three high-level user interfaces (web for casual browsing, programmatic for user-developed clients and commandline for shell access), a middleware layer to manage the distributed components of the systems and hide complexity from the user, and the backend resources (databases, disks, etc) that would be shared by the services. The Python *Flask*¹ micro-service framework was adopted to implement middleware as a collection of RESTful web services, allowing the client layer to focus on user interface issues and keeping the compute-intensive aspects on the data center servers. The astronomer-friendly interfaces custom to Data Lab are built on top of standard VO protocols and services to allow external access to data, storage and compute resources within Data Lab. These same interfaces can be used to access external VO services by use of a *profile* configuration, and similarly Data Lab services are available to VO-only clients.

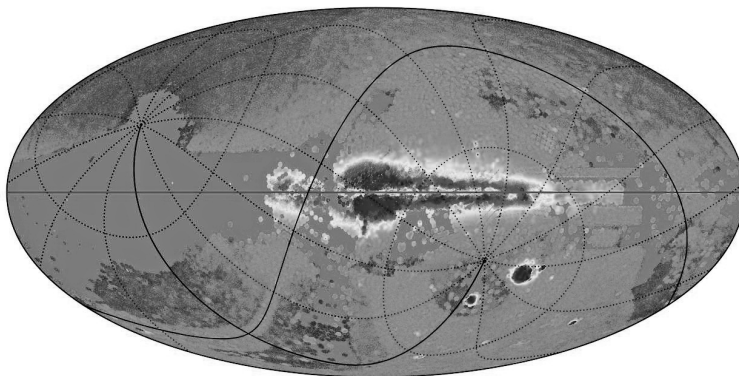


Figure 1. NOAO Source Catalog object density map. Image is mapped to the Log of the number of objects per degree-squared. Catalogs contain 2.9 billion sources and 30 billion measurements. *P.I: D. Nidever*

For more information on this Figure, see Nidever et al. (2018).

¹<http://flask.pocoo.org/>

3. Current Datasets and Capabilities

Data currently available includes >35TB of catalog data, 2PB of image (raw, reduced and coadd) data, and >500M files from survey data releases. These represent a mix of NOAO survey programs, reference data mirrored for efficiency, catalogs and spectra not available from the archive, and datasets published on behalf of external groups.^{2 3} Major datasets include:

Large Surveys:	Dark Energy Survey, DECam Legacy Survey
NOAO Surveys:	SMASH, DECam Plane Survey, DECam Asteroid Database
Reference Data:	Gaia, SDSS, AllWISE, 2MASS, USNO
Original Data:	NOAO Source Catalog, precomputed crossmatch tables
Published Data:	PHAT, S-PLUS, LSST Simulation Catalog

Although some aspects of Data Lab are still under development, current capabilities include:

Multiple Interfaces:	Web, commandline, Python API, Jupyter
Authentication:	Custom token-based authentication + anonymous use
Virtual Storage:	File storage (<i>VOSpace</i>) plus personal database (<i>MyDB</i>)
Sky Exploration:	Survey discovery tool (<i>Aladin lite</i>), catalog overlay (web)
Catalog Query:	SQL queries from web, API, commandline or VO Clients
Image Query:	Scriptable image query/access
File Access:	File-based access to select survey datasets
Visualization:	Web, notebooks, python API tools
Analysis:	Jupyter notebooks environment (relevant packages avail.)

4. Data Publication and Sharing

The NOAO Science Archive is primarily an image repository, however the final data products of a given program often include catalogs, publications, master calibration data and even software, items that cannot be easily searched using traditional archives. With the variety of catalog, image and file-based query/access services, Data Lab has the capability to utilize or deliver the full range of data products to the community, even when downloading for local analysis is not possible due to the size of the collection. A number of non-NOAO datasets are currently being hosted as we explore the requirements for providing a simple path for users wishing to publish their own results using the same web services provided to hosted datasets. Similarly, we wish allow users to configure these same web services for either personal or shared use until final results can be published for general use by the community.

Existing public data collections, e.g. SDSS DR14, may be accessed using the public *File Service* feature of *Virtual Storage*. Users additionally are able to mark areas of their own storage as “*public*” to share file-based data. *Group* (user-defined)

²Southern Photometric Local Universe Survey: <http://www.splus.iag.usp.br/en/>

³Panchromatic Hubble Andromeda Treasury: <https://datalab.noao.edu/phat/index.php>

and *Resource* management will be further developed alongside *Compute Services* to allow users to create and manage catalog and image access services equivalent to those provided by Data Lab.

5. Community Development

Having access to shared analysis tools, e.g. Jupyter notebooks or compute tasks, is just as important as access to the data itself. Compute services will be developed using Docker⁴ and we plan to create a repository for user-built containers that other users can import into their workflow. Likewise, we plan to provide means for users to publish example, tutorial, or analysis notebooks to aide others in getting started with using Data Lab.

We are actively working with educators interested in using Data Lab as a teaching tool as well as other science platforms developers to find interoperable solutions to common problems. All code will be made available to the Open-Source community through GitHub as new tools and services are released.

6. Lessons Learned

Our goal was to span the range of user expertise, science use-cases and data variety, but to balance those with the finite compute resources available. The approach of providing a mix of web-based, programmatic and commandline interfaces to the platform achieves much of that goal, however priorities for which interface is a priority shift rapidly and can be challenging. Similarly, as our user base grows the focus on new development versus feature enhancements and maintenance work must also change. Starting with a user-centric design and evolving that into a user-centric operations model can be difficult but is necessary for success.

References

Nidever, D., et al. 2018, *Astronomical Journal*, 156, 131. [arXiv:1805.02671](https://arxiv.org/abs/1805.02671)

⁴A Linux virtualization technology that allows applications to run in isolated *containers* for enterprise deployment. See <http://docker.com>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Astropy and the Virtual Observatory

Tom Donaldson

Space Telescope Science Institute, Baltimore, MD, USA; tdonaldson@stsci.edu

Abstract. The International Virtual Observatory Alliance (IVOA) has been defining standards for interoperable astronomical data exchange since 2002. Many of these standards are being used successfully and extensively by archives and end user tools to enable data discovery and access. Nevertheless a skepticism persists in parts of the community about the utility and even relevance of these standards, as well as the processes by which they were written. By contrast, the Astropy Project, with its very different processes (and somewhat different goals), has been widely embraced by the community for the usefulness and usability of its interoperable Python packages. In this talk I will discuss what these projects might learn from each other, and how more collaboration might benefit both projects and the community in general.

1. What is the Virtual Observatory?

The vision of the Virtual Observatory (VO) is that astronomical datasets, tools and services should work seamlessly together. The International Virtual Observatory Alliance (IVOA) was formed in 2002 as a partnership of national organizations to further the vision of the VO. The IVOA provides a framework for discussing and sharing VO ideas, seeking input from, and communicating to, the astronomical community. Within that framework, the technical standards needed to make the VO possible are debated and formalized. (Quinn et al. 2004)

2. VO Actively in Use

Many popular tools make use of the VO for data discovery and access, including TOPCAT (Taylor 2005), Aladin (Boch et al. 2007; Bonnarel et al. 2000), the MAST Discovery Portal (Donaldson et al. 2012), ESA Sky (Giordano et al. 2018), WorldWide Telescope (Rosenfield et al. 2018) and DS9 (Joye & Mandel 2003). Large and small archives around the world provide data via IVOA data services, which are discoverable via IVOA standard registries. In fact, there are thousands of distinct data services currently registered by dozens of distinct data publishers (Araya et al. 2015). Access to the ESA Gaia Archive (Salgado et al. 2017) was built on the IVOA Table Access Protocol (TAP) (Dowler et al. 2010).

3. Challenges for the VO

In spite of that significant presence within the community, the VO faces important perception problems that limit progress. The IVOA is seen as a somewhat insular organization, not always driven by community needs. Standards are created and evolve painfully slowly. With standards that have been adopted, lingering usability issues lead to some frustration by developers and users alike. And, more broadly, the relationships and intended workflows among the many standards are not always clear.

While these criticisms are sometimes overstated, they are also fair to an important degree. They result in some potential users and developers being less willing to engage with the VO, thus limiting progress towards its vision of interoperability.

4. Potential Synergies with Astropy/Astroquery

The Astropy project was started in 2011 "as a largely community-driven effort to standardize core functionality for astronomical software in Python." (Astropy Collaboration et al. 2013, 2018) Utilizing an open-source and open-development model, it has evolved into a feature-rich core library, and an ecosystem of affiliated packages. One such package is Astroquery (Ginsburg et al. 2018), which is a set of tools for querying astronomical web forms and databases. Both Astropy and Astroquery have become very popular among end user astronomers.

Astroquery and the IVOA have similar and overlapping goals regarding data discovery in that they both want to provide a predictable way to query astronomy data sets. Astroquery achieves this through a client-side collection of Python modules, one per data provider or collection, which follow a consistent query pattern. The IVOA approach defines the consistency on the server side through a collection of data discovery protocols which then can be predictably queried by client-side software.

The Astroquery approach to predictable discovery lowers the bar of entry for client programmers, because the details of the individual queries are encapsulated behind the Astroquery wrappers. The VO approach, though more complex for a client developer, allows access from any programming language and includes programmatically-readable metadata describing some properties of query results. By adding wrappers for VO queries to Astroquery, users (who are already comfortable with Astroquery) would have predictable access to many more data resources, all without needing to add separate Astroquery modules for each resource.

More importantly, such exposure to VO services could bring a wider audience, and open-development processes, to VO standards. Since those processes have demonstrably resulted in usable Astropy/Astroquery interfaces that address concrete astronomy use cases, they could also help evolve VO standards to be more usable and driven by community needs. While the VO could benefit from this wider audience, Astropy/Astroquery could also benefit by having a larger community actively involved in developing those packages.

The VO could also benefit from Astropy work already done. For example, the IVOA is currently exploring potential standards for data models for various astronomical concepts such as sky coordinates. Learning from related models that already exist (and therefor are already used) in Astropy, could help the IVOA ensure that their models are useful, and indeed compatible with existing Astropy use cases.

5. VO Already in Astropy

The notion of including VO functionality in Astropy is not new. Astropy already includes a very robust VOTable parser. Astroquery includes a VO Cone Search module and utilities for accessing TAP services which are used by multiple Astroquery modules including Gaia (Segovia 2016). In addition, Astropy has another affiliated package called PyVO which contains a number of utilities and wrappers for working with IVOA protocols. Although PyVO has many useful features, it was never well-integrated with Astropy workflows, perhaps because it includes assumptions that users are familiar with certain VO jargon and protocols. This lack of integrated use cases lead PyVO to not see much community involvement since its initial development.

6. Conclusions

The existing VO components in Astropy leave plenty of room for more robust VO features and use cases. Interesting design questions may come up about where such features belong within the Astropy ecosystem (Astropy core, Astroquery, PyVO, other?). Fortunately the open-development model supplies a framework for such questions to be discussed and decided. VO features will be added only when the community agrees that they are worthwhile, in scope, and well-developed. Then that same feedback can inform the IVOA's development of the standards themselves, hopefully increasing the utility of the standards and buy-in from the community at large.

References

- Araya, M., Solar, M., & Antognini, J. 2015, *New Astronomy*, 39, 46 . URL <http://www.sciencedirect.com/science/article/pii/S1384107615000251>
- Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., Ardelean, C., Babej, T., Bach, Y. P., Bachetti, M., Bakanov, A. V., Bamford, S. P., Barentsen, G., Barmby, P., Baumbach, A., Berry, K. L., Biscani, F., Boquien, M., Bostroem, K. A., Bouma, L. G., Brammer, G. B., Bray, E. M., Breytenbach, H., Buddelmeijer, H., Burke, D. J., Calderone, G., Cano Rodríguez, J. L., Cara, M., Cardoso, J. V. M., Cheedella, S., Copin, Y., Corrales, L., Crichton, D., D'Avella, D., Deil, C., Depagne, É., Dietrich, J. P., Donath, A., Droettboom, M., Earl, N., Erben, T., Fabbro, S., Ferreira, L. A., Finethy, T., Fox, R. T., Garrison, L. H., Gibbons, S. L. J., Goldstein, D. A., Gommers, R., Greco, J. P., Greenfield, P., Groener, A. M., Grollier, F., Hagen, A., Hirst, P., Homeier, D., Horton, A. J., Hosseinzadeh, G., Hu, L., Hunkeler, J. S., Ivezić, Ž., Jain, A., Jenness, T., Kanarek, G., Kendrew, S., Kern, N. S., Kerzendorf, W. E., Khvalko, A., King, J., Kirkby, D., Kulkarni, A. M., Kumar, A., Lee, A., Lenz, D., Littlefair, S. P., Ma, Z., Macleod, D. M., Mastropietro, M., McCully, C., Montagnac, S., Morris, B. M., Mueller, M., Mumford, S. J., Muna, D., Murphy, N. A., Nelson, S., Nguyen, G. H., Ninan, J. P., Nöthe, M., Ogaz, S., Oh, S., Parejko, J. K., Parley, N., Pascual, S., Patil, R., Patil, A. A., Plunkett, A. L., Prochaska, J. X., Rastogi, T., Reddy Janga, V., Sabater, J., Sakurikar, P., Seifert, M., Sherbert, L. E., Sherwood-Taylor, H., Shih, A. Y., Sick, J., Silbiger, M. T., Singanamalla, S., Singer, L. P., Sladen, P. H., Sooley, K. A., Sornarajah, S., Streicher, O., Teuben, P., Thomas, S. W., Tremblay, G. R., Turner, J. E. H., Terrón, V., van Kerkwijk, M. H., de la Vega, A., Watkins, L. L., Weaver, B. A., Whitmore, J. B., Woillez, J., Zabalza, V., & Astropy Contributors 2018, *AJ*, 156, 123. 1801.02634

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., Unther, H. M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J. E. H., Singer, L., Fox, R., Weaver, B. A., Zabalza, V., Edwards, Z. I., Azalee Bostroem, K., Burke, D. J., Casey, A. R., Crawford, S. M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P. L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., & Streicher, O. 2013, *A&A*, 558, A33. 1307.6212
- Boch, T., Fernique, P., Bonnarel, F., Allen, M. G., Bienaymé, O., & Derrière, S. 2007, *Highlights of Astronomy*, 14, 625
- Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. 2000, *Astronomy and Astrophysics Supplement Series*, 143, 33
- Donaldson, T., Rogers, A., & Wallace, G. 2012, in *Astronomical Data Analysis Software and Systems XXI*, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461, 327
- Dowler, P., Rixon, G., & Tody, D. 2010, *Table Access Protocol Version 1.0*, Tech. rep.
- Ginsburg, A., Sipocz, B., Parikh, M., Woillez, J., Groener, A., Liedtke, S., Robitaille, T., Deil, C., Norman, H., Svoboda, B., Brasseur, C. E., Tollerud, E., Persson, M. V., Seguin-Charbonneau, L., Armstrong, C., de Val-Borro, M., Morris, B. M., Mirocha, J., Yadav, A., Seifert, M., Droettboom, M., Moolekamp, F., James Allen, Bostroem, A., Egeland, R., Singer, L., Rol, E., & Grollier, F. 2018, *astropy/astroquery*: v0.3.7 release. URL <https://doi.org/10.5281/zenodo.1160627>
- Giordano, F., Racero, E., Norman, H., Vallés, R., Merín, B., Baines, D., López-Caniego, M., Martí, B. L., de Teodoro, P., Salgado, J., Sarmiento, M. H., Gutiérrez-Sánchez, R., Prieto, R., Lorca, A., Alberola, S., Valtchanov, I., de Marchi, G., Álvarez, R., & Arviset, C. 2018, *Astronomy and Computing*, 24, 97
- Joye, W. A., & Mandel, E. 2003, in *Astronomical Data Analysis Software and Systems XII*, edited by H. E. Payne, R. I. Jedrzejewski, & R. N. Hook, vol. 295, 489
- Quinn, P. J., Barnes, D. G., Csabai, I., Cui, C., Genova, F., Hanisch, B., Kembhavi, A., Kim, S. C., Lawrence, A., Malkov, O., Ohishi, M., Pasian, F., Schade, D., & Voges, W. 2004, vol. 5493, 5493. URL <https://doi.org/10.1117/12.551247>
- Rosenfield, P., Fay, J., Gilchrist, R. K., Cui, C., Weigel, A. D., Robitaille, T., Otor, O. J., & Goodman, A. 2018, *The Astrophysical Journal Supplement Series*, 236, 22
- Salgado, J., González-Núñez, J., Gutiérrez-Sánchez, R., Segovia, J. C., Durán, J., Hernández, J. L., & Arviset, C. 2017, *Astronomy and Computing*, 21, 22. 1710.10509
- Segovia, J. C. 2016, *Tap/tap+*. <https://astroquery.readthedocs.io/en/latest/utis/tap.html>
- Taylor, M. B. 2005, in *Astronomical Data Analysis Software and Systems XIV*, edited by P. Shopbell, M. Britton, & R. Ebert, vol. 347, 29

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Lilith: A Versatile Instrument and All-Sky Simulator and its Application to TESS

Jeffrey C. Smith,^{1,2} Peter Tenenbaum,^{1,2} Jon M. Jenkins,² and Joseph D. Twicken^{1,2}

¹*SETI Institute, Mountain View, CA, USA; jeffrey.smith@nasa.gov*

²*NASA Ames Research Center, Moffett Field, CA, USA*

Abstract. To help facilitate the development of the Transiting Exoplanet Survey Satellite (*TESS*) data analysis pipeline, it was necessary to produce simulated flight data with sufficient fidelity and volume to exercise all the capabilities of the pipeline in an integrated way. As a generator of simulated flight data, *Lilith*, was developed for this purpose. Full instrumental and astrophysical ground truth is available and can be used as a training set for *TESS* data analysis software. Our intention is to continue to tune *Lilith* as real *TESS* flight data becomes available, allowing for an up-to-date simulated set of data products to complement the mission flight data products, thereby aiding researchers as they continue to adapt their tools to the *TESS* data streams.

1. Simulating TESS Data

The Transiting Exoplanet Survey Satellite (*TESS*) (Ricker et al. 2014) is a space-based planet-finding NASA mission. *TESS* will identify the best nearby small ($< 4R_e$) planets for detailed follow-up and characterization. This is accomplished by conducting an all-sky transit survey of $\sim 200,000$ stars within ~ 200 lightyears. For each 27.4-day period (Sector), *TESS* will observe a 24° by 96° swath of sky extending from near the ecliptic equator to the ecliptic pole. *TESS* is expected to discover $\sim 1,000$ small planets less than four times the size of Earth. At least fifty of these small worlds will orbit stars sufficiently bright to allow for determination of their masses. These new worlds' proximity to Earth will enable easy follow-up observation and characterization, such as with the James Webb Space Telescope.

The *TESS* Science Processing Operations Center (SPOC) team at NASA Ames Research Center adapted the Kepler Science Processing Pipeline (Jenkins 2017) for use with *TESS* (Jenkins et al. 2016). The pipeline runs on the NAS Pleiades supercomputer and provides calibrated pixels, simple and systematic error-corrected aperture photometry, and centroid locations for all target stars observed, along with associated uncertainties. The *TESS* pipeline searches through all light curves for evidence of periodic transit signals that occur when a planet crosses the disk of its host star. It generates a suite of diagnostic metrics for each transit-like signature discovered, and extracts

planetary parameters. The results of the transit search will be archived to the Mikulski Archive for Space Telescopes (MAST)¹.

To help facilitate the development of the SPOC science pipeline, it was necessary to produce simulated flight data with sufficient fidelity and volume to exercise all the capabilities of the pipeline in an integrated way. Using a physics-based *TESS* instrument and sky model, *Lilith* creates a set of raw *TESS* data which includes models for the CCDs, readout electronics, camera optics, behavior of the attitude control system (ACS), spacecraft orbit, spacecraft jitter and the sky, including zodiacal light, and the *TESS* Input Catalog (TIC). The model also incorporates realistic instances of stellar astrophysics, including stellar variability, eclipsing binaries, background eclipsing binaries, transiting planets and diffuse light. This data can then be passed to the SPOC pipeline providing full integration tests of the science processing all the way from raw pixel calibration to the generation of archivable data products.

The fundamental process of *Lilith* star field rendering is a mathematical model of a point source that incorporates all the effects that contribute to its rendered appearance. This model is an extension of the Point Spread Function (PSF), and is known as a Pixel Response Function (PRF). The details of the rendering is beyond the scope of this paper but to summarize, we first generate an astrophysics-based model of the star field, we then use physics-based models of the instrument optics, CCD and readout electronics to generate a sub-cadence PRF model and finally pass the astrophysical scene through the PRF to record the pixel values.

2. The Importance of Realistic Simulated Data

The *TESS* pipeline, being a *pipeline*, means the inputs to one component are the outputs of a previous component. It is therefore necessary to produce a data set that introduces all the test features for the entire processing into the raw pixel data. In this way, it is possible to see the effect of interactions between the elements. This, then, requires an integrated simulation: one in which all phenomena from the pixel level (readout noise, etc.) to the astrophysical level (transit signatures, etc.) are generated.

Our intention is to continue to improve *Lilith* as real *TESS* flight data becomes available, allowing for an up-to-date simulated set of data products to complement the mission flight data products, thereby aiding researchers as they continue to adapt their tools to the *TESS* data streams. This will be analogous to the numerous completeness studies which have been performed with the Kepler data set to measure planet detection completeness and determine how well various astrophysical signals can be reliably extracted (Christiansen et al. 2016; Burke et al. 2015). Crucial to these studies is a clear understanding of ground truth. For space-based all-sky surveys, such as *TESS*, there are rarely sufficient independent data streams to provide a ground truth reference. We are therefore compelled to provide realistic simulated data which when passed through the *TESS* pipeline can be used to quantitatively measure signal reliability.

¹<https://archive.stsci.edu/tess/>

3. Generating Training Data for Machine Learning

Full instrumental and astrophysical ground truth is generated for each *Lilith* run and can be used as a training set for *TESS* data analysis software. This use has already been exercised when *Lilith* simulated data was used to train a machine learning classifier for planet candidates (Ansdell et al. 2018). Large planets with deep transit signals are trivial to detect in most cases, but transit signals near the noise floor in the data are quite obscure and only through careful calibration, data reduction and phase-folding methods can the signals under study rise above the noise. Crucial to classifying these detected signals is large quantities of simulated data with a wide distribution of transit signals, astrophysics and instrumental noise.

The studies up till now with *Lilith* have used full simulated data sets where raw data is generated and then fully processed through the *TESS* pipeline. This results in the most realistic and thorough data set but the large processing time to fully run the pipeline limits the quantities of data available. The data can be easily multiplied with slight augmentations, such as injecting large numbers of transits on each generated light curve, quickly increasing the data set. Or when real flight data is available, artificial transits can be injected on the real data, such as with the completeness studies performed with Kepler cited above. There are advantages to both augmentation methods. Using real flight data with injected transits will best represent the astrophysical and instrumental noise in the data. However, the ground truth of the real flight data is not entirely known, but fully simulated data has the advantage where the whole data set can be used to train our methods with the full knowledge of the signals. We can then identify, for example, which instrumental or astrophysical signals are contributing most to the reduction in accuracy of a classifier.

4. Performance

The goal is to generate simulated data sets that precisely mirror real flight data. *Lilith* simulations so far were conducted before real *TESS* flight data was available. The best simulated data set was a 4-sector run where 16,000 targets per sector were generated using the TIC 6 catalog. Preliminary candidate target lists (CTLs) were used to select targets and a current SPOC 3.0 pipeline version was used. As of the writing of this paper, the SPOC has processed 3 Sectors of real flight data and so we can begin to do direct comparisons. Figure 1 gives two methods to compare the quality of the light curves. Red data points is simulated and blue is real. The *left* figure shows the Combined Differential Photometric Precision (CDPP) for a 1-hour transit duration in parts per million (ppm), which is a measurement of the noise characteristics of the data and how easily a transit-like signal can be detected (lower CDPP is better). It is plotted versus target stellar magnitude. The sharp discontinuity in target density at magnitude 8 is merely target selection. There are two principle signal characteristics to be gleamed in the figure. *Firstly*, the spread in the data gives the distribution of stellar variability in the light curves. We can see the *Lilith* data has more spread than the Sector-2 data, which implies the simulated data set has more stellar variability. Tuning *Lilith* to include lower average stellar variability amplitudes would help to bring the two data sets in line. *Secondly*, the lower bound to the scatter gives the instrumental noise in the data (I.e. in the limit of quiet stars, the remaining noise is instrumental). To aid the eye, the solid curves are the moving 10th percentiles for each data set. We can see that the

solid curves are in reasonable agreement, implying the instrumental noise in the *Lilith* data mirrors the flight data moderately well. A slight tuning of the amplitude of *Lilith* instrumental noise sources should bring the two data sets in line.

The *right* figure shows the singular values after performing Singular Value Decomposition (SVD) on each data set. The singular values give a numerical estimate of the complexity of the signals in the data. If the singular values drop off quickly then a small number of singular vectors can be used to represent the majority of the signals in the data set. We can see that the Sector-2 data has higher singular values, implying the complexity of signals are greater in the real flight data. We clearly need to introduce more sources of noise in *Lilith*. White Gaussian Noise is good for some noise but colored noise is also necessary. For example, observations have shown that the spacecraft pointing jitter will vary over a sector and should be better simulated. This added complexity will help “shift” the red curve to the right.

Considering the simulated data set was generated before real flight data was available we achieved rather well agreement. A small number of adjustments to *Lilith* configuration parameters will result in a convergence of the data. Beyond simple tuning, methods such as genetic algorithm-based optimization or Generative Adversarial Networks (GANs) could be employed to help tune the simulator to better represent the real data. We intend to investigate these advanced methods in the future.

The authors greatly appreciate the contributions of the full SPOC team in making the *TESS* pipeline a success. We wish to thank Eric Ting, in particular, for his dedication to SPOC operations.

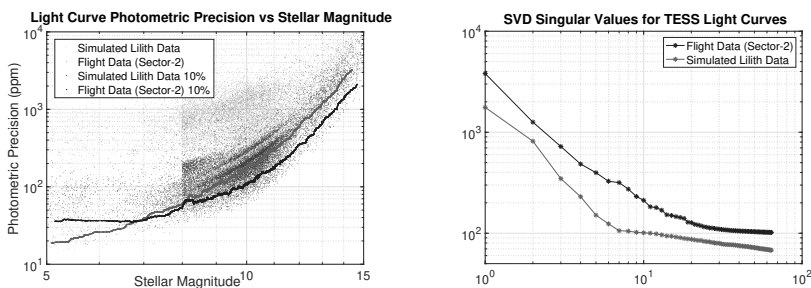


Figure 1. A comparison of simulated to real data light curves *Left*: Comparison of photometric precision. The solid lines are moving 10th percentiles *Right*: Comparison of singular values after performing SVD on the two light curve data sets.

References

- Ansdell, M., et al. 2018, ArXiv e-prints. 1810.13434
- Burke, C. J., et al. 2015, ApJ, 809, 8. 1506.04175
- Christiansen, J. L., et al. 2016, ApJ, 828, 99. 1605.05729
- Jenkins, J. M. 2017, Kepler Data Processing Handbook: Overview of the Science Operations Center, Tech. rep., NASA
- Jenkins, J. M., et al. 2016, in Software and Cyberinfrastructure for Astronomy IV, vol. 9913 of SPIE, 99133E
- Ricker, G. R., et al. 2014, in Space Telescopes and Instrumentation 2014: Optical, Infrared, and Millimeter Wave, vol. 9143 of SPIE, 914320. 1406.0151

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

AFLAK: Visual Programming Environment with Quick Feedback Loop, Tuned for Multi-Spectral Astrophysical Observations

Malik Olivier Boussejra,¹ Shunya Takekawa,² Rikuo Uchiki,¹
Kazuya Matsubayashi,³ Yuriko Takeshima,⁴ Makoto Uemura,⁵ and
Issei Fujishiro¹

¹*Keio University, Yokohama, Kanagawa, Japan; malik@boussejra.com*

²*Nobeyama Radio Observatory, Minamimaki, Nagano, Japan*

³*Kyoto University, Asakuchi, Okayama, Japan*

⁴*Tokyo University of Technology, Hachioji, Tokyo, Japan*

⁵*Hiroshima University, Higashi-Hiroshima, Hiroshima, Japan*

Abstract. This paper describes a free (as in freedom), extendable graphical framework, `aflak`, that provides a visualization environment in particular for astrophysical data set. By leveraging visual programming techniques via an approach based on a node editor, `aflak` allows the busy astronomer to conduct fine-grained processing on multi-spectral data sets. While connecting compute nodes together in the node editor, the final output of the transformations is smoothly displayed in a dedicated visualization window. This enables the astronomer to fine-tune all the interactive parameters of their program with a direct feedback loop.

1. Introduction

Astrophysics is a domain of knowledge where precision and reproducibility are absolute. Mostly, the only data astronomers are able to gather from far objects is their light. And from this light they must create, confirm or invalidate theories via the careful analysis of many case studies. Visually interacting with the data not only assists the astronomer in finding particular objects, but it also helps in the design of programs to verify the relevance of the computing by smoothly and regularly checking the output.

In this paper, we present a free and open-source software framework, `aflak` (Advanced Framework for Learning Astrophysical Knowledge), which is mainly aimed at dynamically analyzing multi-dimensional data. `aflak` can load a data set, and provide a visual programming paradigm to apply transformations on it and visualize their outputs in real time, thus providing a fast and smooth feedback loop to astronomers. `aflak`, with its built-in support for FITS files and astrophysical processing, is currently especially adapted for multi-spectral astrophysical data.

2. Related Works

Astrophysics has had many viewers for FITS files. Most of these tools are free and open-source software. One of the most famous and most used viewer is SAOImage DS9 by Joye & Mandel (2003), which can open FITS files and offer basic analytic needs. Lately, QFitsView (Ott 2012) has been gaining tractions. Even some commercial endeavors, such as NightLight, have been released by Muna (2017). However, the previously mentioned tools are mainly viewers and do not offer many features for data analytics. Data analytics is mostly conducted with other tools, the oldest of which being IRAF (Tody 1986), then supplanted by PyRAF (De La Pena et al. 2001). IRAF, through PyRAF, paved their ways to Astropy, a Python library that can tackle most of the computing needs of astrophysicists (Robitaille et al. 2013) (e.g. transformation and image algebra).

Now, as stated above, there is currently a clear separation between tools for viewing and analyzing in the astronomy ecosystem. Astrophysicists have a workflow consisting in manually analyzing data sets by applying and composing transformations on them. Only then do they export the result, e.g. as a FITS file, to visualize it inside a viewer. Even for Astropy, external tools (e.g. `matplotlib`) are required to view the results. `aflak`'s objective is to provide an integrated environment to both analyze and view astronomical data, with very fast iterations. While `matplotlib` may provide printing quality graphs, it is not really suitable for fast iterations on relatively big data sets. On the other hand, `aflak` (<http://aflak.jp>) allows the user to compose algebraic transforms to implement new nodes using the provided visual programming interface. The user can combine pluggable primitives to create their own macros. More than just allowing to organize nodes, these macros could then be exported and shared among their peers (planned feature). `aflak` provides an image algebra feature similar to that of NumPy, with which the user can play to smoothly visualize the resulting computations. In a word, `aflak` gives fine-grained control through a visual programmatic interface, but with immediate feedback thanks to the integrated data viewer. In the next section, we will present all of `aflak`'s currently implemented features.

3. aflak

`aflak`'s interface is presented in Figure 1. The upper layer of `aflak`'s architecture consists of a node editor engine and a dedicated plotting library to visualize the output data. The node editor engine has a compute back-end, which we called `cake`, that manages pending computational tasks in a multi-threaded manner, decoupled from the UI thread. `aflak` is built from the ground up in *Rust* (<http://www.rust-lang.org>), and is light enough to smoothly cope with gigabyte-sized data sets on a modern but standard laptop in less than a second, enabling true interactivity and responsiveness. *Rust* was chosen for its memory safety that does not sacrifice computing speed, and the relative ease—compared to bare C/C++—of running highly computational tasks on several threads. The user interface is drawn using *OpenGL* via bindings to the *Dear ImGui* Immediate Mode Graphical User Interface library, originally implemented in C++ (<https://github.com/ocornut/imgui>). `aflak` is supported on both *Linux* and *macOS*, and can be ported to other platforms, as all technical choices are portable.

The visual programming interface is composed by a node editor, where one can author a visual program by creating/deleting nodes (of any of three types: value, trans-

formation, and output), and making connections between their input and output slots. Node can be combined to create more complex operations. Besides aflak can easily be extended with new nodes by loading a function implementing the binary interface recognized by aflak to deal with new use cases.

When an output node is created, a corresponding visualization window is spawned, displaying in real time the data that is flowing into this output node. Whenever an error arises during the computing process, a clear error message will be propagated to the visualization window. Moreover, the visualization window provides usual visualization features such as advanced plotting for 1D and 2D data sets. aflak prioritizes dynamic and interactive plotting dealing with fast varying data, contrary to what can be seen in *matplotlib* (Hunter 2007), which can output printing quality plots at the expense of speed and interactivity. New value nodes (node containing a value, not a transformation) can be created and updated from the visualization window, enabling fine-grained control over the value of the node from the output data. Data output can be exported as standard-compliant FITS files containing the history of their generation. Finally, the whole visual program itself can be exported in an *ad hoc* serialization format to guarantee reproducibility of the study. Current aflak has a few built-in examples, such as arbitrary slicing on a 3D dataset or the computing of the equivalent width of a spectral line, from which one can start an analysis or evaluate the software.

4. Concluding Notes

aflak is a nascent project. Many features still need to be included for it to be fully usable by a broad range of astronomers. In order of importance, the wanted features are macro support and batch processing over many different but similar inputs. Implementation of more domain-specific transformations is desirable. In addition, convenience UI functions such as copy-pasting or bulk-selection of nodes are desirable. Interoperability with Virtual Observatory standards, e.g. when loading FITS data from open-access data repositories or when recording the provenance of the output data, are heavily considered.

Acknowledgments. This work is supported by JSPS KAKENHI Grant Numbers 17K00173 and 17H00737.

References

- Bundy, K., et al. 2015, *The Astrophysical Journal*, 798, 7
- De La Pena, M., White, R., & Greenfield, P. 2001, in *Astronomical Data Analysis Software and Systems X*, vol. 238, 59
- Hunter, J. D. 2007, *Computing in Science & Engineering*, 9, 90
- Joye, W., & Mandel, E. 2003, in *Astronomical Data Analysis Software and Systems XII*, vol. 295, 489
- Muna, D. 2017, *Publications of the Astronomical Society of the Pacific*, 129, 058003
- Ott, T. 2012, *Astrophysics Source Code Library*
- Robitaille, T. P., et al. 2013, *Astronomy & Astrophysics*, 558, A33
- Tody, D. 1986, in *Instrumentation in astronomy VI* (International Society for Optics and Photonics), vol. 627, 733

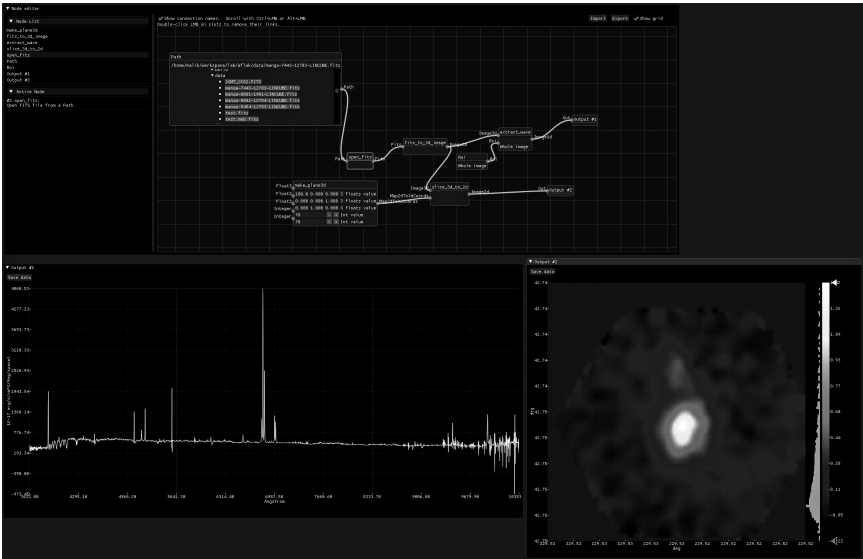
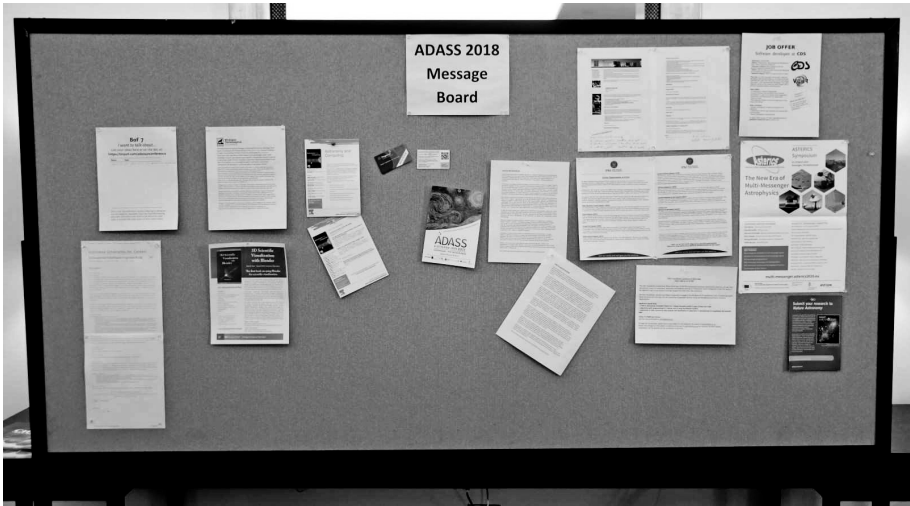


Figure 1. aflak showing a galaxy from the SDSS MaNGA data set by (Bundy et al. 2015). Above, you can see the node editor window. Below, you can see a window for each of the connected output nodes. Each window shows the data that flows into its dedicated output node. All parameters can be updated. All the visualized data depending on the updated parameters will get updated immediately.



Not all ADASS communications are electronic (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

DEAVI: Dynamic Evolution Added Value Interface

Deborah Baines, Ignacio de la Calle, Jose Maria Herrera-Fernandez, Aitor Ibarra, Jesus Salgado, and Luis Valero-Martin

Quasar Science Resources S. L., Edificio Ceudas, Ctra. de La Coruna Km 22.300, 28232, Las Rozas de Madrid, Madrid, Spain;
deborahbaines@quasarsr.com

Abstract. We present DEAVI, an Added Value Interface (AVI) to manage and exploit data from the ESA missions Gaia and Herschel. AVIs are software packages that provide scientists with the mechanisms to submit their own code to be executed close to the ESA mission archives. GAIA AVIs are deployed at the Gaia Added Value Interface Platform (GAVIP), a Python-based platform designed and developed by ESA and hosted at the European Space Astronomy Centre (ESAC). The proposed AVI is part of the software package being developed by Quasar Science Resources for the StarFormMapper (SFM): A Gaia and Herschel Study of the Density Distribution and Evolution of Young Massive Star Clusters project, funded by the European Union under the Horizon 2020 programme.

1. Introduction

The European Space Agency (ESA) operates numerous missions, both operational and scientific. Two of ESA's scientific missions are Gaia¹ and Herschel². The main objective of Gaia is to obtain a three-dimensional map of the Milky Way, i.e. the positions of the stars of our galaxy and their radial and positional velocity measurements. In the case of Herschel, one of its main goals was to study the formation and evolution of stars and galaxies and their interaction with the interstellar medium. Both missions can be considered a success given the number of discoveries they have made. However, combined, they can give a more complete picture of star formation. The StarFormMapper (SFM)³ project is a European H2020 RIA project that proposes a combined study of the data of both missions covering all stages of star formation, from the formation of molecular cores to the dispersion of gas in young clusters. The SFM consortium is a collaboration between the Universities of Leeds (UK, Coordinator), Cardiff (UK) and Joseph Fourier, Grenoble (FR), as well as the Spanish company Quasar Science Resources, S.L. (QSR).

¹<http://sci.esa.int/gaia/>

²<http://sci.esa.int/herschel/>

³<https://starformmapper.org/>

QSR⁴ is a private company that provides consulting Software and System Engineering services for Research and Development projects. The team includes Computer System Analysts, Software and Data Archive Engineers and Scientists. In this project, QSR is developing the necessary software tools in order to handle the scientific algorithms for the analysis of the combined Gaia, Herschel and other data of young star clusters, including the visualisation of the results. These tools have been collected in the first version of the Dynamic Evolution Added Value Interface (DEAVI) software which is presented in this work.

2. Dynamic Evolution Added Value Interface

The DEAVI developed by QSR consists of a virtual infrastructure that can store and run algorithms, as well as visualise data in 2 and 3 dimensions. In addition, the possibility of data exchange between the client and the server has been incorporated through the implementation of a Simple Application Messaging Protocol (SAMP; Taylor et al. 2015) interface.

The infrastructure has been developed in a Docker container (Merkel 2014), which is a structure that wraps a software module containing all the needed elements to invoke and run the software. Docker containers can be invoked using input parameters and can be integrated in virtualised environments such as Amazon Web Services, Puppet, VMware, etc. Figure 1 shows a simple scheme of the virtual architecture where the main components are highlighted, and are described below.

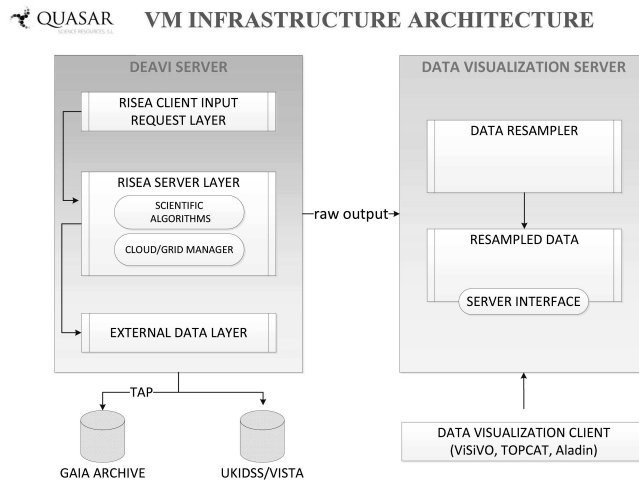


Figure 1. Virtual Server architecture design for the SFM project showing the different subsystems. RISEA server/client stands for Remote Interface to the Stellar Environment Algorithm, and is the main component of the system. TAP stands for Table Access Protocol to access archive data.

⁴<http://www.quasarsr.com/>

2.1. DEAVI server

The DEAVI server has three different layers:

- The first layer is the Remote Interface to the Stellar Environment Algorithm (RISEA) Interface. This is a client interface to access the scientific algorithms, either in production or in development, as designed by the scientific team within the consortium. The Client allows adding, modifying and implementing physical conditions, input parameters and output data.
- The second layer is the RISEA Server which runs on the virtualised infrastructure and is used to handle and inject different algorithms.
- The third layer is the Data Access Interface. It consists of a set of data access mechanisms allowing access to the Gaia and Herschel data, as well as to the auxiliary data used by the algorithms. Herschel data is available for download using the Archive Interoperability interface (HAIO) from the ESA Science Archives. Access to the Gaia catalogue is available using the IVOA Tabular Access Protocol (TAP; Dowler et al. 2011) interface implementation.

2.2. Data visualisation server

The Data visualisation server is the other important part of the virtual architecture. It consists of two components: a Data Resampler that resamples the data on the server side to allow client visualisation, and a Data Visualisation Client that allows the exploration of the results in 3D. Thanks to this part of the software, scientists can visualise data in the client and interact with them using the Bokeh, Astropy and D3.js libraries. As an example, Figure 2 shows a simple algorithm that obtains, from the Gaia catalogue, the positions of a number of stars in a cluster and plots them together with arrows, representing their proper motions. In this prototype, the graphical user interface implements the following sections:

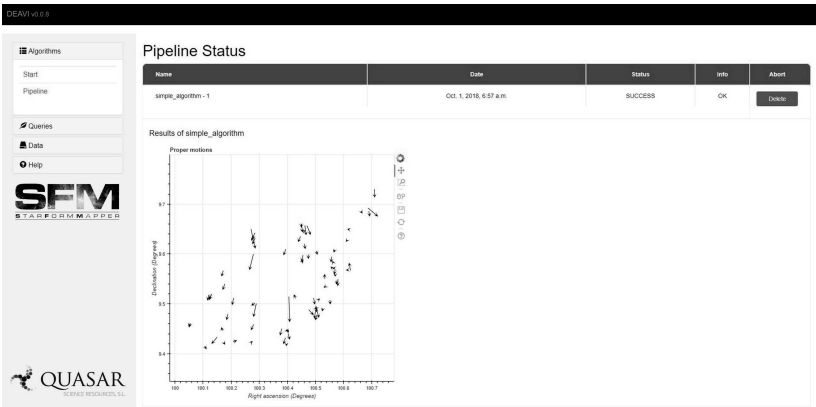


Figure 2. Example of a simple algorithm in DEAVI: star positions with their proper motions. The libraries used to create the plot are Astropy and Bokeh.

- **Algorithms.** This section is dedicated to the execution of the algorithms developed by the scientific team of the consortium. Within this section it is possible to choose and execute an algorithm, visualise its status and the results.
- **Queries.** To speed up the process of making queries to the scientific archives a friendly interface has been implemented to perform queries on GAIA and Herschel data. The results of these queries can then be fed as input to the available algorithms.
- **Data.** This section is where the different stored user data is displayed.
- **Help.** This last section is dedicated to providing the necessary information to the user about the use and functionalities of DEAVI.

3. Deployment of DEAVI in GAVIP

The concept of exploitation platforms has gained prominence over the past few years. The main purpose of these platforms is to offer the possibility of executing algorithms close to the data when the data to be explored is massive. An example of these platforms is the Gaia Added Value Interface Platform (GAVIP; Vagg et al. 2016). GAVIP is a Python-based platform that allows the global scientific community to run scientific code. It is installed at the European Space Astronomy Centre (ESAC) where the data of the missions Gaia and Herschel are stored. Taking into account both factors, we have decided to deploy our software in GAVIP. This adds two important advantages: a) allows data processing without moving the mission data through the network; b) more computing power thanks to the use of the ESAC infrastructure (RAM, HDD, etc.).

4. Conclusions

The StarFormMapper is a project funded by the Horizon 2020 program of the European Union for the study of massive star and star cluster formation. SFM combines data from two of ESA's major space missions, Gaia and Herschel. Quasar Science Resources has developed a value-added interface capable of working simultaneously with both data sets. This interface allows scientists to add new algorithms and visualise original data and results in a friendly and intuitive way. The first version of the software has been deployed on the GAVIP platform at ESAC. Since the data of the Gaia and Herschel missions are also at ESAC, the movement of data through the network is avoided.

Acknowledgments. This work was supported by the SFM project which receives funding from the European Union's Horizon 2020 Research and Innovation Action (RIA) programme under Grant Agreement No 687528.

References

- Dowler, P., Rixon, G., & Tody, D. 2011, ArXiv e-prints. 1110.0497
 Merkel, D. 2014, Linux J.
 Taylor, M. B., Boch, T., & Taylor, J. 2015, ArXiv e-prints. 1501.01139
 Vagg, D., O'Callaghan, D., McBreen, S., Hanlon, L., Lynn, D., & O'Mullane, W. 2016, ArXiv e-prints. 1605.09287

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

New Python Developments to Access CDS Services

Matthieu Baumann and Thomas Boch

CNRS Observatoire de Strasbourg, Strasbourg, Alsace, France;
matthieu.baumann@astro.unistra.fr

Abstract. We will present recent developments made in the frame of the ASTERICS project and aimed at providing Python interface to CDS services and Virtual Observatory standards. Special care has been taken to integrate these developments into the existing `astroquery` environment.

A new `astroquery.cds` module allows one to retrieve image or catalog datasets available in a given region of the sky described by a MOC (Multi Order Coverage map) object. Datasets can also be filtered through additional constraints on their metadata.

The `MOCpy` library has been upgraded: performance has been greatly improved, unit tests and continuous integration have been added, and the integration of the core code into the `astroquery.regions` module is under way. We have also added an experimental support for creation and manipulation of T-MOCs which describe the temporal coverage of a data collection.

1. Python Packages Presentation

1.1. `MOCpy` (Boch & Baumann 2015): a Library Handling the Creation and Manipulation of MOCs

New features and improvements have been added to the library:

- `MOCpy` (Boch & Baumann 2015) has been optimized and tends to use `numpy`'s broadcasting feature as much as possible. Creating a MOC from a list of `astropy.SkyCoord` is a lot faster thanks to the vectorization involved when operations are directly done on `numpy` arrays.

The following code shows the implementation of `from_lonlat` responsible for creating a MOC from lon and lat `astropy` quantities at a given order. This code:

- Uses `astropy-healpix` to get the HEALPix cells where the (lon, lat) coordinates are located.
- Build a $N \times 2$ `numpy` array storing the intervals of the HEALPix cells at a given order.

No Python loops over the quantities are involved here as it is encouraged to perform operations directly on `numpy` arrays.

- Dependencies to `healpy` have been removed. We now use `astropy-healpix` and therefore have changed the license of `MOCpy` (Boch & Baumann 2015) from GPL to BSD-3.

- A new `serialize` method has been added, taking an optional `format` argument that can be set to `fits` or `json`.
- New methods `fill` and `perimeter` have been implemented. These methods are responsible for plotting the MOC (resp. its perimeter) on a matplotlib axis using a projection defined by an `astropy.wcs.WCS` object.

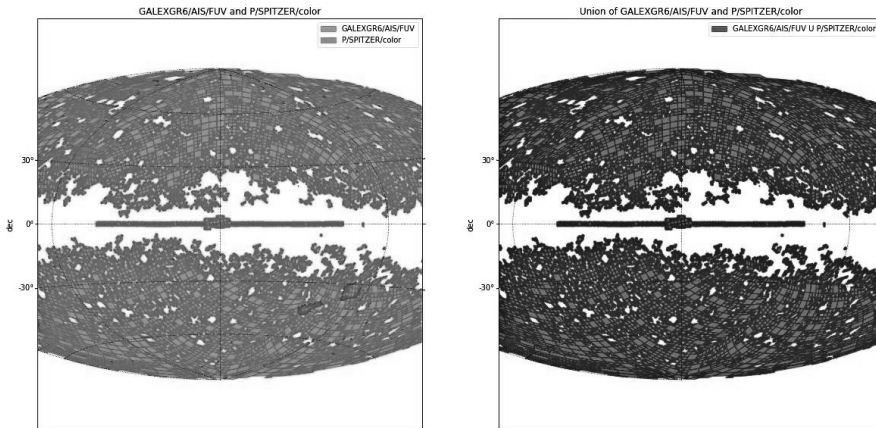


Figure 1. Union of the MOCs between GALEXGR6/AIS/FUV and SPITZER

- A new `TMOC` class handles the creation and manipulation of temporal MOCs. A `from_times` method creates a T-MOC object from an `astropy.time.Time` object. As for the spatial MOCs, it is possible to `serialize` a T-MOC, compute the intersection, union, difference between several T-MOCs as well as use them to filter an `astropy.time.Time` object.

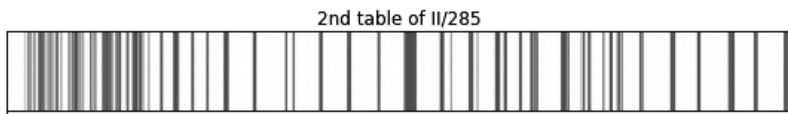


Figure 2. Example of a T-MOC created from II/285

First observation: 1978-05-10 20:09:28.672

Last observation: 2004-04-22 16:56:36.350

Total duration: 227.424 jd

Max order: 14

1.2. `astroquery.cds` (Baumann 2018a): a New Module for Retrieving Data Collections Based on Region and/or Meta-data Queries

`astroquery.cds` (Baumann 2018a) has been merged into the master branch of `astroquery` in July the 23th and will be available for its next release (v0.3.9). This module requests

the CDS MOCServer, a server storing MOCs and meta-data of $\simeq 20000$ data collections. This package offers two methods (see the module’s documentation (Baumann 2018a) for more details):

- `query_region` retrieves the collections having their observations in a specific region. Regions can be expressed as `mocpy.MOC` objects, circle or polygon sky regions.
- `find_datasets` retrieves the collections based on a constraint on their meta-data.

These two methods return by default an `astropy.table.Table` containing the meta-data of one collection per row. An optional argument `return_moc=True` can be used to directly retrieve the MOC (a `mocpy.MOC` object) of the matching collections.

Below 1 is an example of an `astropy` table returned by `query_region` and filtered to select only the vizier tables having between 75000 and 100000 sources. The meta-data shown here are `obs_id`, `obs_title` and `dataproduuct_type`. For a list of all the possible meta-data returned by the `cds` module, please refer to the page 18 of the HiPS IVOA paper (Fernique et al. 2017).

Table 1. Example of an `astropy` table returned by `astroquery.cds`

obs_id	obs_title	dataproduuct_type
I/208/ppm3	The 90000 stars Supplement to the PPM Catalogue (Roeser+, 1994) (ppm3)	catalog
I/237/catalog	The Washington Visual Double Star Catalog, 1996.0 (Worley+, 1996) (catalog)	catalog
I/276/catalog	Tycho Double Star Catalogue (TDSC) (Fabricius+ 2002) (catalog)	catalog

2. State of the Art of the CDS Python Tools

The following image 3 results from a notebook (Baumann 2018b) combining different Python packages, most of them being developed by the CDS team through the past years. It is available on the `cds-astro` github repository as an example for astronomers. This script:

1. Retrieves two MOCs from the MOCServer (Baumann 2018a).
2. Computes their intersection (Boch & Baumann 2015) and shows the resulting MOC on an `aladin-lite` view (`ipyaladin`).
3. Searches for a vizier table in optical regime having some observations in this region (Baumann 2018a).
4. Retrieves the table using `astroquery.vizier`.

5. Filters the table to only keep the observations lying in the MOC (Boch & Baumann 2015) and adds the filtered table to the aladin view (ipyaladin).



Figure 3. Aladin-lite view showing a Vizier table filtered by a MOC

3. Future Improvements

- MOCpy (Boch & Baumann 2015) is currently being integrated into astropy-regions. New classes, `MOCskyRegion` and `MOCpixelRegion` will be implemented. `MOCskyRegion` is the equivalent of the `mocpy.MOC` class, therefore it will contain all its features (serialization, intersection, ...). A `MOCpixelRegion` is a MOC sky region projected using an astropy WCS object.
- `query_region` from `astroquery.cds` will be upgraded to accept `MOCskyRegion` objects.
- The `query_region` methods of both `astroquery.Simbad` and `Vizier` should accept `MOCskyRegion` too so that Simbad and Vizier tables can be filtered by MOCs.

References

- Baumann, M. 2018a, `astroquery.cds` documentation page, <https://astroquery.readthedocs.io/en/latest/cds/cds.html>
- 2018b, Notebook example illustrating the state of the art of the CDS Python tools, <https://github.com/cds-astro/ADASS-IV0A18>
- Boch, T., & Baumann, M. 2015, Python library to easily create and manipulate MOCs (Multi-Order Coverage maps), <https://github.com/cds-astro/mocpy>
- Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017, HiPS - Hierarchical Progressive Survey Version 1.0, IVOA Recommendation 19 May 2017. 1708.09704

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Breathing New Life into an Old Pipeline: Precision Radial Velocity Spectra of TESS Exoplanet Candidates

G. B. Berriman¹, D. Ciardi², B. J. Fulton², J. C. Good², M. Kong², H. Isaacson³, and J. Walawender⁴

¹*Caltech/IPAC-NExScI, Pasadena, CA 91125, USA; gbb@ipac.caltech.edu*

²*Caltech/IPAC-NExScI, Pasadena, CA 91125, USA*

³*University of California, Berkeley, CA 94720, USA*

⁴*W. M. Keck Observatory, 65-1120 Mamalahoa Hwy, Kamuela, HI 96743, USA*

Abstract. The High Resolution Echelle Spectrograph (HIRES) at the W.M. Keck Observatory (WMKO) is one of the most effective Precision Radial Velocity (PRV) machines available to U.S. astronomers, and will play a major role in radial-velocity follow-up observations of the tens of thousands of exoplanets expected to be discovered by the Transiting Exoplanet Sky Survey (TESS) mission. To support this community effort, the California Planet Search (CPS) team (Andrew Howard, PI) has made available a PRV reduction pipeline that will be available to all U.S. astronomers from February 2019 onwards. Operation of the pipeline has strict requirements on the manner in which observations are acquired, and these will be fully documented for users at the telescope.

The pipeline is written in IDL, and was developed over time for internal use by the CPS team in their local processing environment. Development of a modern version of this pipeline in Python is outside the scope of our resources, but it has been updated to support processing in a generic operations environment (e.g. changes to support multiple simultaneous users). We have developed a modern, Python interface to this updated pipeline, which will be accessible as a remote service hosted behind a firewall at the NASA Exoplanet Science Institute (NExScI). Users will be able to use Python clients to access data for input to the pipeline through the Keck Observatory Archive (KOA). The pipeline will create calibrated and extracted 1D spectra and publication-ready time series, which can be visualized and analyzed on the client side using tools already available in Python. The Python client functions interface with the pipeline through a series of server-side web services. Users will have access to a workspace that will store reduced data and will remain active for the lifetime of the project. This design supports both reduction of data from a single night or long-term orbital monitoring campaigns.

1. Introduction

There has been no public service for the reduction of Precision Radial Velocity (PRV) spectra acquired with the Keck/High Resolution Echelle Spectrograph (HIRES), which uses an iodine cell to derive a stable and accurate wavelength calibration. A PRV pipeline developed in IDL and maintained the California Search (CPS) has enabled exoplanet discovery with Keck/HIRES data since 1995 (Marcy & Butler 1992; Butler

et al. 1996; Howard et al. 2010). The CPS has generously made this pipeline available as a public service to enable the determination of masses of exoplanets expected to be discovered by the Transiting Exoplanet Sky Survey (TESS). Starting in 2019, it will be available to Keck Principal Investigators (PIs) for processing of newly acquired data that are acquired with the HIRES instrument in a prescribed fashion and archived at the Keck Observatory Archive (KOA).

The design and performance of the public service is the subject of this paper. The design was driven by two considerations. One was that there were too few resources to rewrite the pipeline in a modern language such as Python. The other was that the pipeline is unsuitable for release as Open Source, because it requires an IDL license, because the results are sensitive to IDL version and machine architecture, and because the highest accuracy radial velocities are achieved by building a persistent database of reference stars for wavelength calibration. The public service will, therefore, be operated on a dedicated server at NExScI, and users will interact with it through a Python client library.

2. System Architecture

The server side environment is shown in the upper left panel of Figure 1, and the IDL code has been modified to operate in it. Users are allocated a permanent workspace to store data processed by the pipeline. Access to the pipeline is through Python clients, which connect to the operations environment through CGI services, which themselves log on users with KOA-issued credentials, invoke data reduction, manage the processing, and extract PRVs. The upper right panel of Figure 1 shows these interfaces, and the lower left panel shows the detailed design of the Python API. In operations, the pipeline is seamlessly interoperable with KOA: once users access the pipeline, raw data for the program to be processed are automatically transferred to the workspace and reduced to spectra for each order. Users then invoke, and control, the creation of radial velocity calculations. Processing a full night's observations will generally take no more than 5 to 7 hours.

3. Pipeline performance

The lower right panel of Figure 1 shows the performance of the pipeline through a graph of the photon-limited single measurement precision as a function of exposure meter setting and signal-to-noise ratio. Signal-to-noise ratios below 70 (shaded gray) are not officially supported by the pipeline and may produce erratic results.

4. Science Verification

We have conducted a program of science verification by comparing results from the pipeline with those in the literature. Figure 2 shows one example, HD 7924. This star hosts three exoplanets; planet b published in 2009 with 7 years of data (198 RVs); planets c and d published in 2016 with 18 years of Keck data and 1.5 years of APF data, for a total of 907 RVs. By processing 167 measurements made in 2010, the NExScI pipeline is able to recover these three planets. Table 1 compares the NExScI radial velocity semi-amplitudes with those in the literature (Fulton et al. 2015).

Table 1. Comparison of the radial velocity semi-amplitudes (m/s) of HD 7924

Exoplanet	Fulton et al. (2015)	NExSci Pipeline
b	3.6	3.55
c	2.3	1.7
d	1.7	2.7

Acknowledgments. The PRV pipeline is a collaboration between the NASA Exoplanet Science Institute, the Keck Observatory Archive, the California Planet Survey and the W. M. Keck Observatory. We thank the many contributors who have developed the IDL pipeline over the past 25+ years, including: Paul Butler, Geoff Marcy, Jeff Valenti, Steve Vogt, Debra Fischer, Andrew Howard, Jason Wright, John Johnson, Chris McCarthy, Eric Williams, Howard Isaacson, and B.J. Fulton.

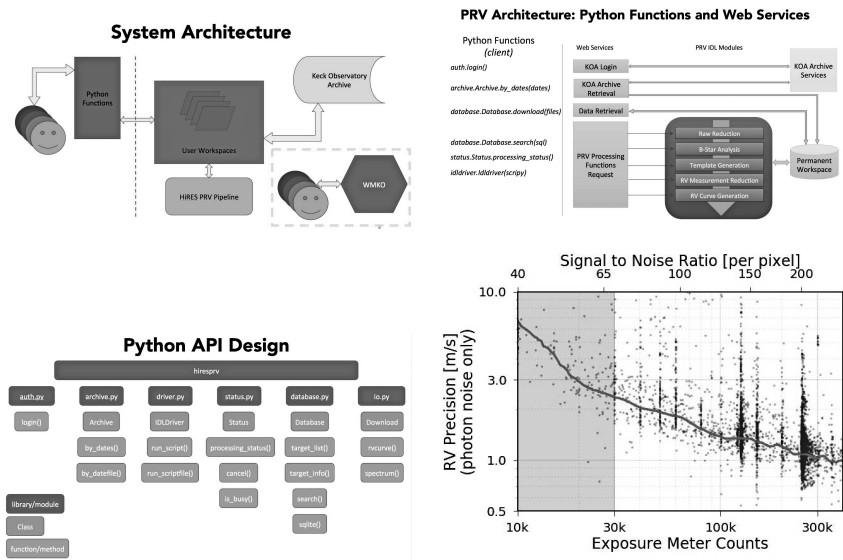


Figure 1. Upper Left: The System Architecture; Upper Right: Python Interfaces to the PRV; Lower Left: The Python API; Lower Right: Photon-limited single measurement precision as a function of exposure meter setting and signal-to-noise ratio. The red line traces the lower 30th percentile of the individual measurements (blue points)

References

Butler, R. P., Marcy, G. W., Williams, E., McCarthy, C., Dosanji, P., & Vogt, S. S. 1996, PASP, 108, 500
Fulton, B. J., Weiss, L. M., Sinukoff, E., Isaacson, H., Howard, A. W., Marcy, G. W., Henry, G. W., Holden, B. P., & Kibrick, R. I. 2015, ApJ, 805, 175

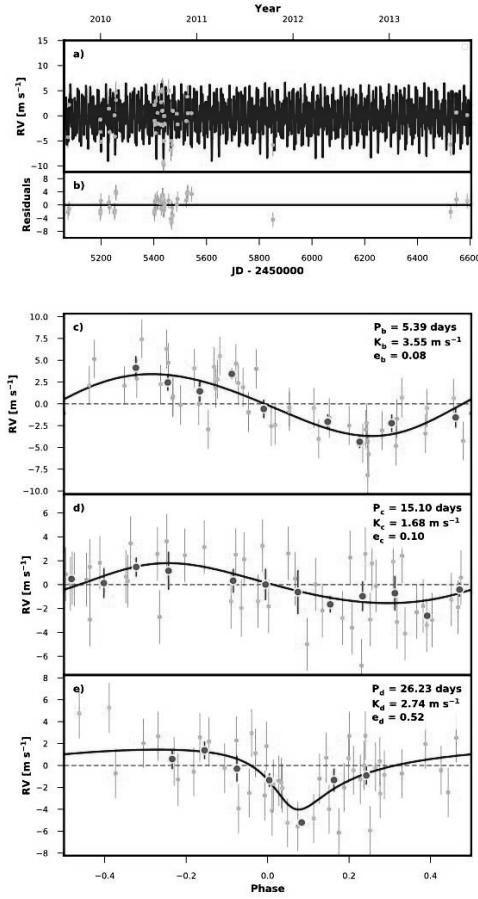


Figure 2. Recovery of the three known exoplanets in HD 7024. The Upper panel shows all the measured radial velocities for this star. The lower panels show the recovery of the three exoplanets by applying the NExScI pipeline to data acquired in 2010 (large circles), cf. those in the literature (small circles)

Howard, A. W., Johnson, J. A., Marcy, G. W., Fischer, D. A., Wright, J. T., Bernat, D., Henry, G. W., Peek, K. M. G., Isaacson, H., Apps, K., Endl, M., Cochran, W. D., Valenti, J. A., Anderson, J., & Piskunov, N. E. 2010, *ApJ*, 721, 1467. 1003.3488
 Marcy, G. W., & Butler, R. P. 1992, *PASP*, 104, 270

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Gaia Photometric Science Alerts Data Flow

A. Delgado, S. Hodgkin, D.W. Evans, D.L. Harrison, G. Rixon,
F. van Leeuwen, M. van Leeuwen and A. Yoldas

Institute of Astronomy, University of Cambridge, Cambridge, UK
mdelgado@ast.cam.ac.uk

Abstract. The Gaia Photometric Science Alerts project started operations in 2014 and will continue until 2020, with the goal of discovering transient events in the Gaia data. Since the mission began, more than 6000 transients have been discovered or had their existence confirmed. The data is received from the spacecraft and processed by the Alerts pipeline and flows through different web applications for selecting and publishing the transient alerts. After the publication of the findings, the Gaia Marshall web application serves as a meeting point for astronomers to interact for prioritizing and optimizing the follow-up and classification of the alerts.

This paper is a brief overview of the Gaia Photometric Science Alerts data flow, from the alert candidate extraction to their follow-up and classification.

1. Alerts candidates discovery and selection

Since Gaia was launched in 2014, the data has been transmitted daily from the spacecraft, and received and processed by the Gaia Photometric Science Alerts System (see fig. 1) at the Institute of Astronomy (Cambridge, UK).

The Gaia Photometric Alerts pipeline, AlertPipe, processes the data received daily and extracts potential new sources and objects increasing or decreasing in brightness.

After data processing, detection of transients and filtering of those transients to remove data artifacts, the surviving transient event candidates (alerts) are visually inspected using a web application, known as the Eyeballing App, which displays the Gaia data as well as ancillary information on each candidate. This application enables the final assessment of the publishability of the alerts.

For every alert, the Eyeballing App shows the data available from Gaia: the spectra, lightcurve, environment plots, Gaia data-release 2 parameters and, when possible, an HR diagram with the candidate superimposed.

Additionally, to allow a visual inspection of the alert's location, the Eyeballing App shows the Aladin Lite (Boch & Fernique 2014) and SDSS finding charts. It also connects to Simbad and NED to learn more about the candidate's location, checks VizieR VSX catalog to discover whether it is already a known transient or variable object, and acquires information from SkyBot (Berthier et al. 2006) to allow the user to reject spurious transients due to data affected by the close proximity of planets and their satellites. The probability of other solar-system objects crossing the field of view is evaluated within AlertPipe using an ephemeris of minor planets generated within Gaia DPAC, and this information is displayed within the Eyeballing App. Also shown in the App, are the results of the cross-match between the alert position and with data

assembled from the hourly parsing of other transient surveys, Astronomer's Telegrams¹ and the Transient Name Server website².

The Eyeballing App allows each user to vote and comment on each alert. When an alert reaches the voting threshold for publication, a link to publish the alert is enabled.

The use of this tool is restricted to the Gaia Alerts team.

2. Alerts publication

After the inspection of the alerts data and the selection of the candidates, the findings are made publicly available to the astronomical community in multiple formats:

- Through a dedicated website³ in CSV, HTML and RSS formats.
- Reported to IAU-Transient Name Server.
- Broadcast as VOEvents using 4 Pi Sky broker (Staley & Fender 2016).
- Notifications to GaiaAlerts mobile apps (iOS and Android).
- Providing data to Gaia in the UK site (<https://gaia.ac.uk>).

The Gaia Photometric Science Alerts web application (Delgado et al. 2019) has a front end with a public area where all the information about the alerts is published, and a restricted area where the administration of the application and the publication of the alert candidates takes place after several online and internal checks.

The back end of the Photometric Science Alerts system manages the updates to the published alerts with the latest data from Gaia once it has been received and processed. The collection of Astronomer's Telegrams and data from various transient surveys available on the Web using ETL (Extract Transform Load) techniques is handled by cron jobs and the parsed data is available through a public interface which allows combining search criteria.

The Gaia alerts catalog can be visualized on the All-Sky interface developed using Aladin Lite (Boch & Fernique 2014) allowing the display of alerts by time ranges and containing the option to focus on the details of a single alert.

3. Gaia Photometric Alerts community: Gaia Marshall

The Gaia Marshall is a password-protected part of the Gaia Photometric Science Alerts web-application. The credentials are granted per user and the access level depends on their assigned group.

The Marshall provides contextual data for the alerts and acts as a place to exchange information and improve the classifications. This web application comprises

¹<http://www.astronomerstelegram.org>

²<https://wis-tns.weizmann.ac.il>

³<http://gsaweb.ast.cam.ac.uk/alerts>

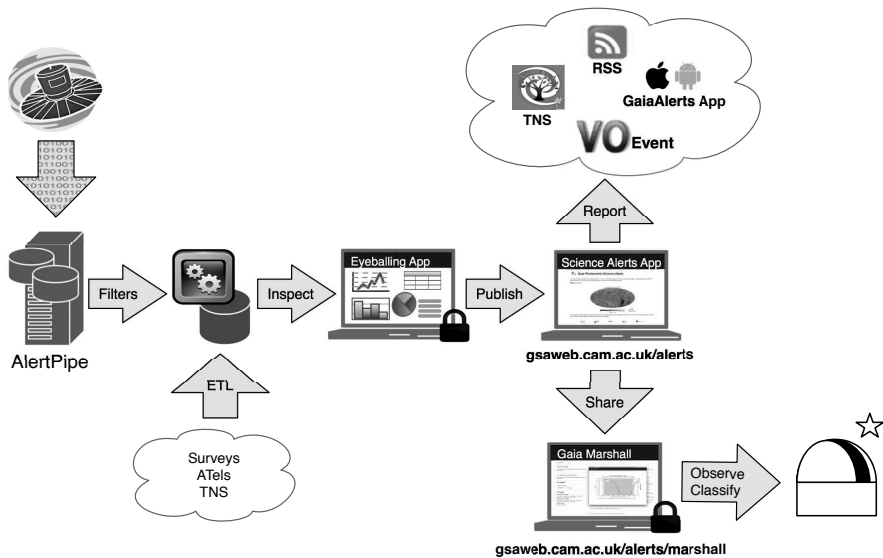


Figure 1. Gaia Science Alerts Data Flow.

the follow-up information, including the data added through the Cambridge Photometric Calibration Server⁴ and the collected data for every alert. It also contains per-alert message boards which support the interaction between astronomers allowing them to prioritize and optimize the follow-up and classification of the alerts.

Each alert has a lifecycle that flows through various states, from New to Active and finally to Archive, if it does not require further follow-up. While an alert is Active, the Marshall users can request actions as Photometry, Spectroscopy or Classify, tagging them as Urgent if that is the case. There are web pages displaying lists of alerts, per-status and per-action, and some of them have counterpart machine-readable files to allow the automation of observations with robotic telescopes. The Status page gives an overview of the Gaia alerts follow-up.

The Marshall has other features, including visibility charts in local and UTC times for the alert, at a range of observatory locations. There is a personalized Favourites List, to which an alert can be added and deleted via the per-alert page or from Favourites List.

The classifications can be entered as suspected or final. The complete history of the alerts are kept and shown to the users to aid in the decision of the final classification, which is only propagated to the public Alerts website and mobile apps when the proposed final classification is accepted by Gaia Alerts team. The follow-up of an alert can continue when the object is classified.

⁴<http://gsaweb.ast.cam.ac.uk/followup>

4. Technologies

The Gaia Photometric Alerts web applications and their backends have been developed in Python Virtual Environments taking advantage of open source astronomical libraries and services, as Astropy (Robitaille et al. 2013), Astroquery (NED, Simbad, Vizier), SkyBot (Berthier et al. 2006), Matplotlib, Bokeh and Comet (Swinbank 2014) among others.

The Eyeballing, Publisher and Marshall web application interfaces have been implemented in Django framework and designed based on the Responsive Web Design (RWD) approach using Bootstrap 3 and Javascript standard libraries like jQuery, jQuery UI, dataTables, Highcharts along with some others tailored specifically for this project.

The database engine used by all these applications is PostgreSQL enhanced with Q3C (Koposov & Bartunov 2006) functionalities.

5. Results

On average, the rate of candidates to be visually selected is 30 per day of which 6 per day are published.

As of today, the classification rate is ~22% of the published alerts: ~75% SNe, ~12% CVs, ~7% AGNs, ~2% YSOs, ~2% ULENSs.

There is work in progress for classifying the Gaia alerts automatically using Machine Learning techniques.

Acknowledgments. This development was supported by the UK Space Agency (ST/N000641/1), Science and Technology Facilities Council (ST/L006553/1) and GENIUS project funded by the European Community's Seventh Framework Programme (FP7-SPACE-2013-1) under grant agreement n° 606740.

References

- Berthier, J., Vachier, F., Thuillot, W., Fernique, P., Ochsenbein, F., Genova, F., Lainey, V., & Arlot, J.-E. 2006, in ADASS XV, edited by C. Gabriel, C. Arviset, D. Ponz, & E. Solano (San Francisco: ASP), vol. 351 of ASP Conf. Ser., 367
- Boch, T., & Fernique, P. 2014, in ADASS XXIII, edited by N. Manset, & P. Forshay (San Francisco: ASP), vol. 485 of ASP Conf. Ser., 277
- Delgado, A., Rixon, G., van Leeuwen, G., Hodgkin, S., Harrison, D. L., van Leeuwen, F., & Yoldas, A. 2019, in ADASS XXVI, edited by M. Molinaro, K. Shortridge, & P. Pasian (San Francisco: ASP), vol. 521 of ASP Conf. Ser., 509
- Koposov, S., & Bartunov, O. 2006, in ADASS XV, edited by C. Gabriel, C. Arviset, D. Ponz, & E. Solano (San Francisco: ASP), vol. 351 of ASP Conf. Ser., 735
- Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., et al. 2013, *Astronomy & Astrophysics*, 558, A33
- Staley, T. D., & Fender, R. 2016, arXiv preprint arXiv:1606.03735
- Swinbank, J. 2014, *Astronomy and Computing*, 7, 12

The CASA Software for Radio Astronomy: Status Update from ADASS 2018

B. Emonts,^{1,2} R. Raba,¹ F. Montesino Pouzols,³ T. Tsutsumi,⁴ T. Nakazato,⁵ A. Kepley,¹ D. Schiebel,¹ S. Castro,³ A. Comrie,⁷ & K.-S. Wang,⁶ (on behalf of CARTA), S. Bhatnagar,⁴ P. Brandt,⁴ C. Brogan,¹ J. Donovan Meyer,¹ P. Ford,⁴ K. Golap,⁴ C.E. García-Dabó,³ B. Garwood,¹ A. Hale,¹ T. Hunter,¹ B.R. Kent,¹ W. Kawasaki,⁵ R. Indebetouw,¹ D. Mehringer,¹ R. Miel,⁵ G. Moellenbrock,⁴ S. Nishie,⁵ J. Ott,⁴ D. Petry,³ M. Pokorny,⁴ U. Rau,⁴ C. Reynolds,¹ K. Sugimoto,⁴ V. Suoranta,¹ N. Schweighart,¹ D. Tafoya,⁵ A. Wells,¹ and I. Yoon¹

¹*NRAO, 520 Edgemont Road, Charlottesville, VA 22903*

²*CASA User Liaison bemonts@nrao.edu*

³*ESO, Karl Schwarzschild Strasse 2, D-85748 Garching, Germany*

⁵*NRAO, 1003 Lopezville Rd, Socorro, NM 87801, USA*

⁴*NAOJ, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, Japan*

⁶*ASIAA, Academia Sinica, PO Box 23-141, Taipei 10617, Taiwan*

⁷*IDIA, University of Cape Town, Rondebosch, 7701, South Africa*

Abstract. CASA, the Common Astronomy Software Applications package, is the primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and the Karl G. Jansky Very Large Array (VLA), and is frequently used also for other radio telescopes. In these proceedings of the 28th Astronomical Data Analysis Software & Systems (ADASS) conference, we give an overview of several new features in the CASA imaging task `TCLEAN`. This includes improved automated masking for image deconvolution, as well as parallel imaging options to increase imaging speeds. In addition, we highlight two upcoming developments. The first is the anticipated arrival of a first-look version of the Cube Analysis and Rendering Tool for Astronomy (CARTA), which is expected to eventually replace the `CASA VIEWER`. The other is a change in the way the different CASA components (e.g., tools and tasks) can be integrated within the Python environment, allowing much greater flexibility for users starting with CASA 6. We also summarize a list of CASA links to guide the user community to the latest CASA information and documentation.

1. Introduction

CASA (McMullin et al. 2007) is being developed with the primary goal of supporting the data reduction and analysis needs of ALMA and the VLA, with a versatility that also benefits the processing of data from other radio telescopes. The CASA package can process both interferometric and single dish data. One of its core functionalities is to support the ALMA, VLA and VLA Sky Survey (VLASS) pipelines.

The CASA infrastructure consists of a set of C++ tools bundled together under an iPython interface as data reduction tasks. This structure provides flexibility to process the data via task interface or as a Python script. In addition, many post-processing tools are available for even more flexibility and special reduction needs.

In these proceedings, we provide a status update of the CASA software and highlight a few upcoming new developments for 2019. We presented these results at the 28th ADASS conference held from 11–15 Nov 2018 at the University of Maryland.

CASA is developed by an international consortium of scientists based at the National Radio Astronomical Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), the Academia Sinica Institute of Astronomy and Astrophysics (ASIAA), the CSIRO division for Astronomy and Space Science (CASS), and the Netherlands Institute for Radio Astronomy (ASTRON), under the guidance of NRAO.

2. tcLEAN: parallelized imaging and automated masking

tcLEAN is the primary CASA task used for imaging and deconvolution (Rau et al in prep).¹ tcLEAN is the successor of CLEAN, which is no longer being actively maintained.

For the first time, the ALMA imaging pipeline (Muders et al. 2014) has the parallel processing mode of tcLEAN enabled during Cycle-6 (CASA 5.4.0), achieving substantial performance improvement (Fig. 1). This is done through MPI (Message Passing Interface) multi-process parallelization (Castro et al. 2017), which speeds up various components of tcLEAN, including gridding, deconvolution, and auto-masking. This parallel mode can be triggered in the mpicasa environment using the parameter `PARALLEL = TRUE` on normal MeasurementSet data. We are confident that this mode utilized by the ALMA imaging pipeline is mature and suitable for general use on most data sets.

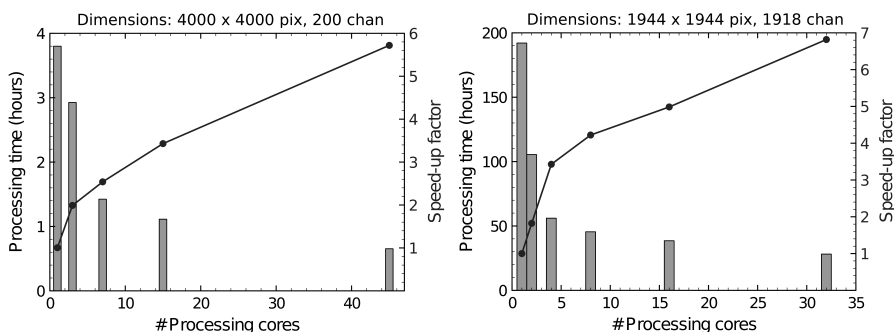


Figure 1. Parallel processing in tcLEAN for two ALMA mosaic data sets. Plotted are the processing time (histogram and left axis) and speed-up factor (blue line and right axis) against the number of cores used for processing. The processing was done with CASA 5.4 [5.2], after [before] the May 2018 upgrade of the Lustre file system at the North American Alma Science Center (NAASC) for the left [right] plot. For the left plot, we took the median value over seven runs for each of the five setups. For additional parallel performance results, see Bhatnagar (2015); Emonts (2018).

¹<http://www.aoc.nrao.edu/~rurvashi/ImagingAlgorithmsInCasa/ImagingAlgorithmsInCasa.html>

Another important new feature in `tclean` is the option of automated masking in the deconvolution process (Kepley et al. in prep). This “auto-multithresh” algorithm (automated masking using multiple thresholds) can be enabled within `tclean` by setting `USEMASK = ‘AUTO-MULTITHRESH’`. The algorithm successfully captures emission from point and extended sources. Although it was originally developed for ALMA, the algorithm also works well for a wide variety of data. ADASS contributions by A. Kepley (O.12.1) and Tsutsumi et al. (2019) (P.12.15) provide details.

3. Data visualization: CARTA

For visualizing data products, CASA relies on the `viewer`. However, the increasing sizes of data products will become ever more challenging for current visualization tools.

The Cube Analysis and Rendering Tool for Astronomy (CARTA) is a new image visualization tool designed for ALMA, VLA, and future radio telescopes such as the Square Kilometre Array (SKA). The CARTA architecture is suitable for visualizing large image cubes. A first-look version of CARTA is planned for release end 2018 (Fig. 2). The CASA team plans on CARTA eventually replacing the CASA `viewer`.

CARTA is being developed by a team consisting of the members from the Academia Sinica Institute of Astronomy and Astrophysics (ASIAA) in Taiwan, the National Radio Astronomy Observatory (NRAO) in the US, the University of Alberta in Canada, and the Inter-University Institute for Data Intensive Astronomy (IDIA) in South Africa.

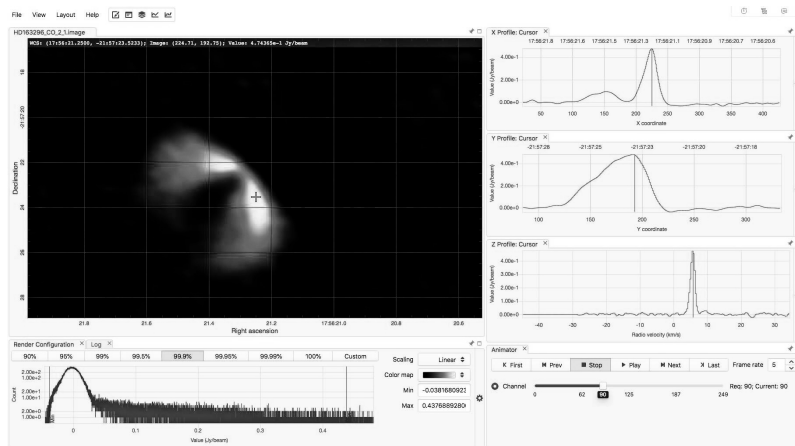


Figure 2. CARTA visualization tool (<http://cartavis.github.io>). Credit: CARTA team.

4. CASA 6: Flexibility in Python

CASA has always been distributed as a single, integrated application, including a Python interpreter and all the libraries, packages and modules. This monolithic distribution makes it difficult to import CASA functionality into existing Python workflows.

As part of a switch from Python 2 to 3, CASA will be reorganized to support building CASA tools with Python’s distutils (and GNU autoconf). This will allow

greater flexibility for users to integrate CASA in their preferred environment, with tools and tasks as standard Python modules (Fig. 3). The CASA group will also continue to produce an all-inclusive distribution for users who are not interested in this new option. The first release under this new approach will be CASA 6.0, expected in mid 2019.

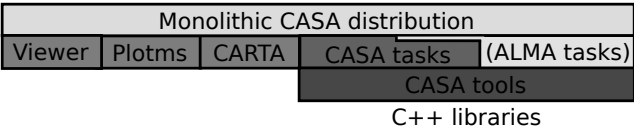


Figure 3. Block diagram of CASA 6. Each block depends on those below, and can be distributed independently. The green GUIs on the left do not depend on Python.

5. CASA Resources

CASA website: please visit for all CASA information: <https://casa.nrao.edu/>

CASA help: for problems or questions, contact the NRAO or ALMA Helpdesk: https://casa.nrao.edu/help_desk_all.shtml

CASA documentation: CASA Docs is the official CASA documentation: <https://casa.nrao.edu/casadocs>

CASA Newsletter: a CASA Newsletter is sent to the community twice a year: https://science.nrao.edu/enews/casa_007

CASA email lists: stay up-to-date on CASA announcements and register: https://casa.nrao.edu/mail_list.shtml

CASA feedback: the CASA team welcomes feedback: casa-feedback@nrao.edu

Acknowledgments. We thank the CARTA and Pipeline teams, and ADASS organizers. The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

References

Bhatnagar, B. 2015, and the CASA HPC Team, CASA Memo 4, CASA Imager Parallelization²
 Castro, S., et al. 2017, in ADASS XXV, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of ASP Conf. Series, 595
 Emonts, B. 2018, CASA memo 5, CASA performance on Lustre: serial vs parallel and comparison with AIPS²
 McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in ADASS XVI, edited by R. A. Shaw, F. Hill, & D. J. Bell, vol. 376 of ASP Conf. Series, 127
 Muders, D., et al. 2014, in ADASS XXIII, edited by N. Manset, & P. Forshay, vol. 485 of ASP Conf. Series, 383
 Tsutsumi, T., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 587

²CASA Memos: <https://casa.nrao.edu/casadocs-devel/stable/knowledgebase-and-memos/casa-memos>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Data Analysis Tools for JWST and Beyond

Henry Ferguson¹ and the JWST Data Analysis Tools Development Forum

¹*Space Telescope Science Institute, Institution City, State/Province, Country;*
ferguson@stsci.edu

Abstract. Data Analysis tools are essential for transforming data into knowledge. They are distinct from pipeline tools in that they usually require interaction and tailoring to specific scientific needs, but parts of the analysis process can often be turned into a pipeline after an initial exploratory phase. Data analysis tools for JWST are being added to the open-source Python+Astropy ecosystem, either as complete packages or as contributions to new packages. The tools include libraries to manipulate and transport complex geometric transformations (gWCS and ASDF), libraries for image analysis (imexam and photutils) and tools for analyzing spectroscopy (Specviz, MOSviz and Cubeviz). We provide a brief summary of the tools, highlighting areas that are ripe for collaboration.

1. Introduction

Astronomers receiving their pipeline-processed JWST data will need to inspect and analyze it. The tasks involved range from fairly common to science specific. Workflows often involve iteration, interaction with the data, and writing custom scripts. Common elements of these workflows involves such generic tasks as: converting, combining, measuring, modeling, and visualizing. The goal of the JWST data-analysis tool development is to provide the building-blocks for such tasks in the python ecosystem.

2. Goals

With the limited lifetime of JWST, it is a high priority to equip astronomers with flexible software tools to analyze and interpret JWST data efficiently right from the start of the mission. Because there is a wide range of science, the strategy is to build a flexible general-purpose library that can be customized to the specific needs of an observing program. The goals are similar to the motivations for building astronomy libraries in IRAF and IDL:

- To reduce the necessity for writing software;
- to make it much easier to write software when necessary;
- to provide a rich library of tools upon which to build;
- to simplify sharing and code re-use;
- and to enhance data sharing between different systems and tools.

It is likely that JWST users will be analyzing data on their local computer as well as migrating more and more to the cloud computing. To date, our visualization tools

have been aimed at local computing. The underlying scriptable libraries, however, will work just as well in the cloud and we plan to provide Jupyter notebooks as tutorials that can be executed in the cloud without requiring installation. Visualization within a JupyterLab environment is progressing rapidly and we anticipate evolving in that direction for the JWST data-analysis tools.

3. The Packages

Stable versions of the software can be installed via AstroConda. Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. AstroConda is a free Conda channel maintained by the Space Telescope Science Institute. This is intended to provide straightforward installation for the entire suite of tools, separated into environments so that they can be installed without interfering with other python installations. The main packages under development are listed in Fig. 1.












Package	Purpose	Maturity
Astropy	A community python library for astronomy	
glue	Linked-dataset exploration; visualization toolkit	
ginga	Image viewer toolkit	
Photutils	Source detection & photometry	
imexam	Interactive image analysis	
astroimtools	convenience tools for working with images	
specviz & specutils	1D spectra analysis	
mosviz	Quick-look analysis for multi-object spectroscopy.	
cubeviz & spectral_cube	3-d spectroscopic analysis	
asdf	Advanced Scientific Data Format (replaces FITS)	
gwcs	Geometric distortion mapping & transformations	

Figure 1. Package maturity as of ADASS2018, using a (non-GMO) fruit-based maturity index. Buds are prototypes with little or no documentation. Cherry pies are ready to be baked into your workflow. We aim to have everything in “cherry pie” state by launch.

The three packages at the top of the list have primarily been developed outside of STScI, with significant contributions from the Institute. The core Astropy package provides a variety of data structures and I/O routines, data-table manipulation, statistics, modeling and fitting, convolution and filtering, some visualization tools, and facilities for dealing with physical constants, units, quantities, coordinates, time and uncertainties. Affiliated packages provide additional functionality; some of the JWST data-analysis tools are being developed as affiliated packages.

Glue is a general-purpose analysis and visualization tool for dealing with linked data-sets. A common astronomical application involves selecting data points on a scat-

ter plot or histogram and highlighting the objects in the subset in an image display. Glue can be used standalone, from python, or from a Jupyter notebook. It can be easily customized and extended.

Ginga is a full featured image viewer and toolkit. Its plugin architecture simplifies customization. It can read data from files or Numpy arrays. It is primarily written and maintained by software engineers at the Subaru Telescope, National Astronomical Observatory of Japan.

Photutils provides tools for detecting and performing photometry of astronomical sources. A recent addition is a python implementation of the Anderson & King (2000) Anderson & King (2000) strategy for building an oversampled "effective PSF" from multiple dithered undersampled images. Imexam is a lightweight library which enables users to explore data from a command line interface, through a Jupyter notebook or through a Jupyter console. It may be used as a replacement for the IRAF imexamine task. It can be used with multiple viewers, such as DS9 or Ginga, or without a viewer as a simple library to make plots and grab quick photometry information. Astroimtools is an astropy affiliated package to provide some commonly used convenience tools for dealing with images. It is a bit of a catch-all for items that didn't fit cleanly in Astropy core or photutils; it is possible that some of the features may migrate into other packages as development progresses.

JWST spectroscopy support centers around two Astropy-affiliated libraries – specutils and spectral-cube – and three visualization tools: specviz, mosviz and cubeviz. Specutils, which is in fairly early development, provides a basic interface for the loading, manipulation, and common forms of analysis of spectroscopic data. These generic data containers and accompanying modules will provide a toolbox that the astronomical community can use to build more domain-specific packages. Spectral-cube provides the equivalent for spectral cubes, optionally with Stokes parameters.

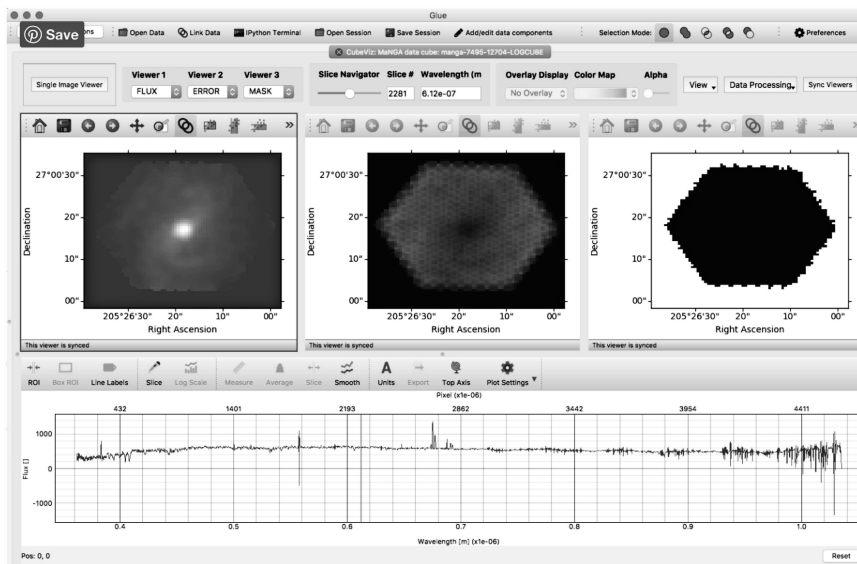


Figure 2. Example of a Cubeviz dashboard viewing data from the MANGA survey.

Specviz is a visualization and interactive data-analysis tool for 1D spectra which supports the most common measurements and workflows of the popular IRAF *splot* task. Mosviz is a glue plugin intended as a quicklook tool for inspecting multi-object spectra. Its dashboard displays the 1D and 2D spectra locked in wavelength and associated cutout images with the slit overlaid locked to the 2D spectrum. Cubeviz (Fig. 2), also a glue plugin, provides full-featured analysis and visualization for IFU data cubes.

The last two packages in the list represent an evolution from FITS and its associated World-Coordinate System conventions to a more capable and more easily extensible framework. The Advanced Scientific Data Format (ASDF) has been designed to facilitate combining hierarchical human-readable metadata with binary data structures in an interchange format that we hope will be convenient for astronomers using JWST and other facilities. JWST data will be packaged in FITS files at least initially, but the complex metadata will be in an ASDF extension. The most common workflows using rectified data will be possible from the FITS extensions alone; those requiring world-coordinate-system transformations will be facilitated by having the transformations fully specified in the ASDF extension.

The goal of the generalized world-coordinate system (GWCS) package is to provide a flexible toolkit for expressing and evaluating transformations between pixel and world coordinates, as well as intermediate frames along the course of this transformation. The GWCS object supports a data model which includes the entire transformation pipeline from input pixel coordinates to world coordinates (and vice versa). The basis of the GWCS object is Astropy modeling. Models that describe the pixel-to-world transformations can be chained, joined or combined with arithmetic operators. This approach allows for easy access to intermediate frames. GWCS provides further transformations between standard celestial coordinate frames. It supports Astropy Quantities and units, which are particularly useful for spectral coordinates. Time coordinates are instances of Astropy Time objects, facilitating conversions and barycentric corrections.

4. Development Strategy

Initial development priorities were guided by a set of about 30 JWST use cases provided by astronomers planning to use JWST. The JWST Science Working Group and Science Advisory Committee have offered input and guidance. Nevertheless, many of the tasks for JWST are similar for other observatories, so the tools should have broad applicability.

Development is all open source, coordinated through Astropy and through the JWST Data Analysis Development Forum (JDADF). We have adopted an Agile development strategy, which is particularly suited to building tools within the rapidly-evolving python ecosystem. Two-week coding sprints facilitate a cycle of rapid development, testing and refinement that is very beneficial for ensuring the tools hit their target.

References

Anderson, J., & King, I. R. 2000, *PASP*, 112, 1360. [astro-ph/0006325](https://doi.org/10.1086/317000)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Optimization Strategies for Running Legacy Codes

Jason Lammers¹ and Peter Teuben²

¹*Computer Science Department, University of Maryland, College Park;*
jllammers@gmail.com

²*Astronomy Department, University of Maryland, College Park*

Abstract. Legacy codes can be both incredibly useful as well as cumbersome. Although they produce robust results from well-defined problems, they plague the user with complex installations, complicated user interfaces, and inflexibilities. Short of rewriting such codes, scripts which emulate the effects of legacy codes can lead to tremendous overhead and more headache for the user. Nonetheless, these legacy codes can provide invaluable insight to a skilled user who takes the time to understand its inner workings.

In this paper, we will showcase examples of the NEMO Stellar Dynamics Toolkit and how analysis simulation codes can be optimized using modern scripting languages. Not only is it possible, it is relatively easy to speed up computation time and increase user-friendliness by integrating modern scripting languages into codes which would otherwise overwhelm an inexperienced user.

1. Introduction

Legacy codes are plentiful in astronomy. Although there are examples of libraries as well, we focus here on applications. The definition of “legacy” can be a bit vague in our field. Apart from the code not being supported anymore (or “orphaned”), it can also mean a code where little or no development takes place because the underlying languages have caused the code to become difficult to maintain (see also the discussion in Portegies Zwart (2018)). The common example in our field is of course FORTRAN, but perhaps also Perl and Tcl/Tk, but there are clear exceptions to this rule.¹

In practice, a number of definitions are commonplace: code inherited by another person or team, code using legacy libraries, code using old compilers etc. But they all have one thing in common: it’s code we still care about, because it has value, but can be in danger of losing it’s livelihood. In rare cases one can resort to virtual machines, as there is now decent support to maintain old software in runtime.

Wrappers are pretty common in modern Unix tools. Classic programs that are rarely used by themselves (e.g. gs) are wrapped in more modern tools (e.g. dvi2pdf).

In the NEMO package (Teuben 1995) a number of these legacy codes are available. They have been proven useful for summerschools, in fact a number of codes

¹ An attempt to count them in ASCL became futile

were written to emulate a paper (e.g. Toomre and Teuben at the PITP 2009 school (priv.comm.) attempting to reproduce the Holmberg (1941) galaxy-galaxy interaction)

In this paper we concentrate on python scripting vs. shell scripting, user interfaces, data I/O and some issues around usability, reproducibility, and sustainability.

2. Scripting

Most codes are not run in isolation, especially when it comes to the analysis portion that is often followed. During research we were interested in the statistical properties of small-N Plummer models to compare with GAIA data in NEMO. Traditionally those scripts were written in the shell (C-shell in our case), and a small 100 sample Monte-Carlo simulation of a 100 body cluster took 30 seconds. It was clear removing a lot of small file I/O overhead could improve the run-time. So the script was rewritten to read the ASCII tables into numpy as soon as NEMO produced them, and the remaining analysis was done using `scipy.stats.binned_statistics` and taking simple mean and std from the accumulated numpy arrays.

This caused the example to run in just 3 seconds, but of course this still meant crossing the boundary from a legacy program (`mkplummer` in our case) into python using an ASCII table and `numpy.loadtxt`. If the analysis would be done in the same memory space as NEMO, it would give an obvious additional boost to the runtime. It so happens that two convenient “classes”, `Moment` and `Grid`, were available in NEMO to do the analysis that we needed. So to reproduce the exact computations done in `csh` and `numpy`, we moved this code in native C within the `mkplummer` task. This caused the runtime to go down to 0.01 seconds!

Another modern example is wrapping executables in python, e.g. the Montage kit as described by Good & Berriman (2019) in this volume. This approach also suffers from file I/O overhead in an extensive pipeline.

3. Compilers

Another problem with legacy code can be the compiler. Not all codes adhere strictly to a language standard. In particular, some “classic” FORTRAN code is now falling victim to the ever-increasing adherence to the FORTRAN standard. A new version of GNU FORTRAN 8.1 caused compilations errors in MIRIAD and an associated RPFITS library, and our code had to be retrofitted.

4. User Interfaces - usability

4.1. Command Line

Legacy codes come with a variety of legacy user interfaces. In an environment such as NEMO this can complicate writing scripts in a uniform way. In addition, FORTRAN based codes often use standard “`fort.N`” files for I/O, complicating running simulations in parallel in the same directory. For this a standard library interface with a run directory was devised. As an example, take the `nbody1` program from Aarseth (2011). One prepares a small ASCII file with parameters, and runs the program as follows

```
% nbody1 < nbody1.in
```


which produces a few dump files, in a direct formatted FORTRAN I/O format. The example with a hand-coded front-end looks as follows:

```
% mkplummer out=plk nbody=1000
% runbody1 in=plk outdir=run1 tcrit=10
% snapplot run1/OUT3.snap
```

where the `runbody1` command runs `nbody1` in a new directory, where the `nbody1.in` file will be located.

4.2. Generalizations (run vs. idf)

The `run` environment in NEMO is a more generic way to run legacy programs. One typically runs the legacy code with an descriptor parameter file, where default parameters have been set. Command line options can then override these parameters, and a new parameter file is written in a run directory to isolate the input and output data in that directory.

```
% run nbody1 nbody1.ini run1 run1.log -- tcrit=10 kz=1,2
% cat nbody1.ini
kstart=1 tcomp=1.0
n=25 nfix=1 nrand=10 nrun=1
eta=0.02 deltat=1.0 tcrit=5.0 qe=2E-5 eps=0.05
kz=1 0 0 0 1 1 0 0 0 0 0 0 0 0
alphas=2.5 body1=5.0 bodyn=1.0
q=0.5 vxrot=0.0 vzrot=0.0 rbar=1.0 zmbar=1.0
```

There are two types of descriptor files: one where the file is verbatim copied, with replaced parameters, and one where the names and values are in two separate files.

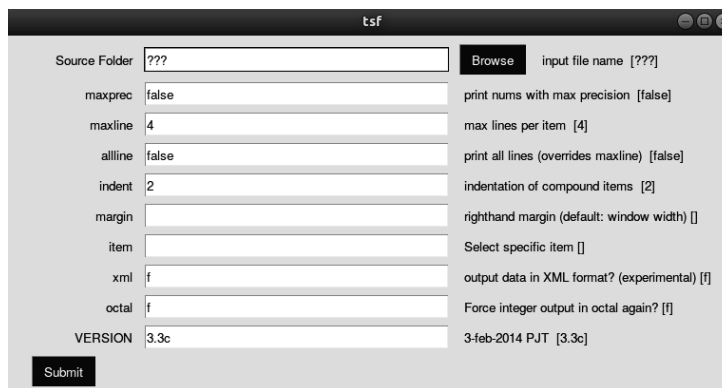


Figure 1. Example self-built GUI for `tsf` using PySimpleGUI

4.3. Graphical User Interfaces

Now that we have a uniform (command) user interface, it can be utilized as a framework for a GUI or even flow diagrams using programs such as Pegasus (Vahi 2019)

and `aflak` (Boussejra 2019). In our case, we used a python package, `PySimpleGUI`,² which was built to interface with, simplify, and generalize `tkinter`'s complex set of settings. We realized we could simply pass commands from the GUI to the command line interface and use the GUI to display the output. In essence, the GUI we made builds itself using the outputs of different NEMO functions.

We used the self-describing `help=h` option to pass arguments and default values. With no modification to any of the original programs we created a simple, friendly way to run the legacy NEMO codes. The `help=z` option can still be used to hook NEMO programs in the `cantata` visual programming interface in `Khoros` (Konstantinides & R. Rasure 1994).³

5. Data I/O

Embedding the user interface can still result in complicated data I/O patterns. For example, in N-body codes it is common that they use their own efficient native language binary format to store the particle positions. But this format is unlikely going to fit into the pipeline of the analysis software. In the case of NEMO's `nbody1`, the input conditions would need to be written in something like a FORTRAN unformatted data stream. The legacy code then reads the data and parameter file, the output files are then decoded back into NEMO's format, after which a NEMO specific pipeline can analyze the data.

An alternative approach is to embed the analysis code inside the run-time system of the N-body code. Using modern plug-in techniques this can be a very modular way to set up such experiments. An example of this can be found in the AMUSE (see e.g. Pelupessy et al. (2013)). Actually, `amuse` is more a hybrid.

References

- Aarseth, S. J. 2011, NBODY Codes: Numerical Simulations of Many-body (N-body) Gravitational Interactions, Astrophysics Source Code Library. 1102.006
- Boussejra, M. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 245
- Good, J., & Berriman, G. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 685
- Holmberg, E. 1941, *ApJ*, 94, 385
- Konstantinides, K., & R. Rasure, J. 1994, Image Processing, *IEEE Transactions on*, 3, 243
- Pelupessy, F. I., van Elteren, A., de Vries, N., McMillan, S. L. W., Drost, N., & Portegies Zwart, S. F. 2013, *A&A*, 557, A84. 1307.3016
- Portegies Zwart, S. 2018, *Science*, 361, 979
- Teuben, P. 1995, in *Astronomical Data Analysis Software and Systems IV*, edited by R. A. Shaw, H. E. Payne, & J. J. E. Hayes, vol. 77 of *Astronomical Society of the Pacific Conference Series*, 398
- Vahi, K. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 689

²<https://github.com/MikeTheWatchGuy/PySimpleGUI>

³there is some irony in the fact that `khoroS` is legacy

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Arcade: An Interactive Science Platform in CANFAR

Brian Major, JJ Kavelaars, Sebastien Fabbro, Daniel Durand, and
 Helena Jeeves

CADC - National Research Council Canada, Victoria, B.C., Canada;
brian.major@nrc-cnrc.gc.ca

Abstract. For a number of years, CANFAR (Canadian Advanced Network for Astronomy Research) has offered virtual machines (VMs) as a way to do both interactive computing and batch processing in the cloud. A VM is a general and flexible base offering that can suit nearly any given astronomy compute project, but demands of users a thorough understanding of the intricacies of software installation and maintenance, and requires significant effort to achieve initial benefits. Arcade is an effort by the Canadian Astronomy Data Centre (CADC) to offer the CANFAR community an astronomy-focused, easier-to-use, and more intuitive science platform for conducting reproducible science.

1. Introduction

Arcade is composed of a number of pre-built, specialized application bundles (Docker¹ containers) that appear in a graphical desktop environment. They run independently of each other in the cloud, allowing Arcade to optimize container execution to their particular resource requirements. Each container has user-specific access to a shared file system and other CANFAR cloud services, outfitting users with a variety of computing, analysis and storage tools.

At the CADC we see a lot of potential in Arcade and will be planning its evolution based on a number of key questions. How can scalability be best achieved? How can we reduce the burden of software and infrastructure maintenance for users and operators? Can we allow users to customize their Arcade experience? How can consoles be launched in batch processing? How can we best leverage open source technology and development from other projects? We shall discuss the core concepts of Arcade and explore its potential in respect to these questions and from feedback from the astronomy computing community.

2. Everything is a Container

Arcade is composed of a number of cooperating containers that expose an API (Application Programming Interface) to users and programs. These cooperating containers are called Arcade's *system containers*. This API to Arcade allows for the creation and management of more containers: *session containers* and *software containers*.

¹<https://www.docker.com/>

Session containers are a hub for software containers. Currently, Arcade offers a single type of session container: a NoVNC² graphical display container. Users connect to these containers with a browser and have the experience of a windowing desktop environment. No other software is available in the NoVNC session container. So, for users to perform tasks, they must launch software containers in the session container.

Software containers must be associated with a specific session container. For NoVNC sessions, the display of the software containers is shown on the NoVNC windowing environment. This makes the software containers appear as though they are running on the session container even though they are not. Any number of software containers can be associated with a session container, limited by the number of resources available in the entire Arcade platform.

Software containers are specialized to performs specific tasks. They are currently being built by the operators of Arcade ‘on demand’, but we envision supporting user contributed containers in the future. Some examples of software containers that have been built are:

- Containers for various versions of CASA (Common Astronomy Software Applications, National Radio Astronomical Observatory McMullin et al. 2007)
- SAOImage DS9 (Smithsonian Astrophysical Observatory³)
- Python software for interacting with CANFAR (VOspace (Graham et al. 2018) client, TAP (Dowler et al. 2010) client, astropy (Price-Whelan et al. 2018))

2.1. Shared Home, Shared Scratch

Each session container and software container that is launched has some common characteristics, one of which is shared storage. The \$HOME directory is the same (per user) which allows containers to share application configuration information and small amounts of data. Also, a larger scratch directory is shared between containers. This directory is intended to be used for staging any temporary files required for processing. Whereas the \$HOME directory will be preserved indefinitely, the scratch directory may be cleared by the system at arbitrary times. These directories are not intended to store processing inputs or outputs. Those products should be stored in one of the CANFAR VOspace implementations where data sharing capabilities exist.

2.2. Shared Credentials

Upon startup, the users’ credentials, in the form of X.509 (Boeyen et al. 2008) proxy certificates, are injected into the container. Any CADC or CANFAR command line tools will make use of these credentials when calling their respective services, thus giving users automatic and individualized authenticated access to those services. Proxy certificates are obtained through the private API of the IVOA Credential Delegation Protocol (Plante et al. 2010).

²<https://novnc.com/>

³<http://ds9.si.edu/>

3. The Future for Arcade

3.1. Scalability

Arcade is currently a prototype running on a single virtual machine. For it to be useful in a production sense (multiple users performing intensive data analysis) it needs to scale appropriately.

The first step in the scalability plan is to use Kubernetes⁴ as a platform for running all three levels of Arcade containers. This transition gives Arcade the ability to scale to the capacities defined in the Kubernetes installation and configuration.

The software containers have different resource utilization requirements. Some may run best with graphical processing units, some may require lots of memory. To satisfy these differences, Arcade can offer a variety of execution environments for the software containers with different resource profiles. When a software container is launched, Arcade can try to match the container to the most appropriate execution environment based on the container's requirements and the current state of the Arcade system. Eventually, Arcade may detect when certain resource profiles are scarce and add more of them to the pool. In this way, users can have a heterogeneous set of software containers running with optimal resource profiles in a single session.

3.2. Custom Containers, Batch Execution

The provided set of software containers in Arcade could be thought of as a starting point for further development. Scientists would launch one (or more) of the software containers that most closely fitted the type of processing being performed. Changes or additions could be made to the container(s) to customize the work until the scientist is satisfied with the results. This is a step towards reproducible science as this container could be referenced in a publication. Thus, we would like to provide a way for users to save this customized version of the container.

If the work is meant to be parameterized and parallelized, the scientist would then like to launch many instances of the container over a set of input parameters. How is this batch execution invoked and defined? These questions will be explored through further prototyping Arcade to meet a core set of batch execution use cases.

References

- Boeyen, S., Santesson, S., Polk, T., Housley, R., Farrell, S., & Cooper, D. 2008, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 5280. URL <https://rfc-editor.org/rfc/rfc5280.txt>
- Dowler, P., Rixon, G., & Tody, D. 2010, Table Access Protocol Version 1.0, IVOA Recommendation 27 March 2010. 1110.0497
- Graham, M., Major, B., Morris, D., Rixon, G., Dowler, P., Schaaff, A., & Tody, D. 2018, VOSpace Version 2.1, IVOA Recommendation 21 June 2018
- McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in ADASS XVI, edited by . D. J. B. R. A. Shaw, F. Hill (San Francisco: ASP), ASP Conf. Ser., 376
- Plante, R., Graham, M., Rixon, G., & Taffoni, G. 2010, IVOA Credential Delegation Protocol Version 1.0, IVOA Recommendation 18 February 2010. 1110.0509

⁴<http://kubernetes.io>

Price-Whelan, A. M., Sip'ocz, B. M., G'unther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., P'erez-Su'arez, D., de Val-Borro, M., Paper Contributors, P., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., Coordination Committee, A., Ardelean, C., Babej, T., Bach, Y. P., Bachetti, M., Bakanov, A. V., Bamford, S. P., Barentsen, G., Barmby, P., Baumbach, A., Berry, K. L., Biscani, F., Boquien, M., Bostroem, K. A., Bouma, L. G., Brammer, G. B., Bray, E. M., Breytenbach, H., Buddelmeijer, H., Burke, D. J., Calderone, G., Cano Rodr'iguez, J. L., Cara, M., Cardoso, J. V. M., Cheedella, S., Copin, Y., Corrales, L., Crichton, D., D extquoterightAvella, D., Deil, C., Depagne, E., Dietrich, J. P., Donath, A., Droettboom, M., Earl, N., Erben, T., Fabbro, S., Ferreira, L. A., Finethy, T., Fox, R. T., Garrison, L. H., Gibbons, S. L. J., Goldstein, D. A., Gommers, R., Greco, J. P., Greenfield, P., Groener, A. M., Grollier, F., Hagen, A., Hirst, P., Homeier, D., Horton, A. J., Hosseinzadeh, G., Hu, L., Hunkeler, J. S., Ivezi'c, Z., Jain, A., Jenness, T., Kanarek, G., Kendrew, S., Kern, N. S., Kerzendorf, W. E., Khvalko, A., King, J., Kirkby, D., Kulkarni, A. M., Kumar, A., Lee, A., Lenz, D., Littlefair, S. P., Ma, Z., Macleod, D. M., Mastropietro, M., McCully, C., Montagnac, S., Morris, B. M., Mueller, M., Mumford, S. J., Muna, D., Murphy, N. A., Nelson, S., Nguyen, G. H., Ninan, J. P., N"othe, M., Ogaz, S., Oh, S., Parejko, J. K., Parley, N., Pascual, S., Patil, R., Patil, A. A., Plunkett, A. L., Prochaska, J. X., Rastogi, T., Reddy Janga, V., Sabater, J., Sakurikar, P., Seifert, M., Sherbert, L. E., Sherwood-Taylor, H., Shih, A. Y., Sick, J., Silbiger, M. T., Singanamalla, S., Singer, L. P., Sladen, P. H., Sooley, K. A., Sornarajah, S., Streicher, O., Teuben, P., Thomas, S. W., Tremblay, G. R., Turner, J. E. H., Terr'on, V., van Kerkwijk, M. H., de la Vega, A., Watkins, L. L., Weaver, B. A., Whitmore, J. B., Woillez, J., Zabalza, V., & Contributors, A. 2018, *aj*, 156, 123



Daniel Durand grabbing some extra ADASS spices after some extra encouragement from Elizabeth Warner (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Exploring Space, Time, and Data with WCSTools

Jessica Mink

Smithsonian Astrophysical Observatory, Cambridge, MA, USA;
jmink@cfa.harvard.edu

Abstract. The WCSTools package, open-source software since 1996, is best known for its ability to set and utilize the world coordinate systems of image data and to search catalogs, but it contains other tasks to observe and manipulate the data and metadata in FITS files. Here are some of them, plus some upcoming additions to the package, including implementations of arbitrary-length keywords in FITS headers and access to additional catalogs.

1. Introduction

The WCSTools package (Mink 2011) was developed in the 1990's as methods of mapping sky coordinates to image pixels started to become standardized, first through the AIPS (Greisen & Cotton 1994) and DSS mappings, and later with the FITS world coordinate system (WCS) standards (Calabretta & Greisen 2002). First came subroutines to implement the WCS (Mink 1996), then catalog and image manipulation subroutines and programs to fit and access image world coordinate systems (Mink 1997), then more programs to make the package more scriptable (Mink 1999). Table 1 displays the current suite of programs, with brief descriptions as produced by the command, `wcstools`.

2. Extending FITS Header Capabilities

The original paper describing the FITS (Flexible Image Transport System) describes a FITS HDU (Header Data Unit), hereafter referred to as a "FITS header", has consisted of "*...one or more logical records giving header information in the form of 80-character ASCII card images, 36 per record. Each card image contains an 8-character keyword (in upper case), usually followed by an equals sign and a value. ... required keywords are followed, in any order, by optional keywords, some of which are described in the FITS papers, and are terminated by an END keyword.*" (Wells et al. 1981).

WCSTools uses FITS headers as random access databases with 80character loosely-formatted records. This turns out to provide a lot of flexibility and makes it possible to rapidly follow the original direction about the random order of non-required keywords.

Table 1. WCSTools 3.9.6 separately callable tasks

Program	Description
addpix	Add a constant value(s) to specified pixel(s)
bincat	Bin a catalog into a FITS image in flux or number
char2sp	Replace this character with spaces in output (default=_)
compix	Operate on all of the pixels of an image
cphead	Copy keyword values between FITS or IRAF images
crlf	Change CR's to newlines in text file (for imwcs, imstar logs)
delhead	Delete specified keywords from FITS or IRAF image file headers
delwcs	Delete the WCS keywords from an image
filename	Drop directory from pathname, returning just the file name
filedir	Drop filename from path name, returning directory path
fileroot	Drop file name extension, returning path name without it
edhead	Edit the header of a FITS or IRAF file
getcol	Extract specified fields from an space-separated ASCII table file
getdate	Convert between two date formats
getfits	Extract portion of a FITS file into a new FITS file, preserving WCS
textbfgethead	Return values for keyword(s) specified after filename
getpix	Return value(s) of specified pixel(s)
gettab	Extract values from tab table data base files
httpget	Send contents returned from URL to standard output
i2f	Read two-dimensional IRAF image file and write FITS image file
imcat	List catalog sources in the area of the sky covered by an image.
imextract	Extract 1D file from 2D file or 2D file from 3D file
imfill	Replace bad pixels in image files with 2-D Gaussian, mean, or median
imhead	Print FITS or IRAF header
immatch	Match catalog and image stars using the WCS in the image file.
imrot	Rotate and/or reflect FITS or IRAF image files
imresize	Block sum or average a file by integral numbers of columns and rows
imsize	Print center and size of image using WCS keywords in header
imsmooth	Filter FITS and IRAF image files with 2-D Gaussian, mean, or median
imstack	Stack 1-dimensional images into a 2-dimensional image
imstar	Find and list stars in an IRAF or FITS image
imwcs	Match FITS or IRAF image stars to catalog stars and fit a WCS
isfits	Return 1 if argument is a FITS file, else 0
isnum	Return 1 if argument is an integer, 2 if it is floating point, else 0
isrange	Return 1 if argument is a range of the format n1[-n2[xs]],...
keyhead	Change keyword names in headers of FITS or IRAF images
newfits	Create blank FITS files (dataless by default with BITPIX=0)
remap	Rebin an image from its current WCS to a new one
scat	Search a source catalog given a region on the sky
sethead	Set header keyword values in FITS or IRAF images
setpix	Set specified pixel(s) to specified value(s)
simpos	Return RA and Dec for object name(s) from SIMBAD, NED, Vizier
sky2xy	Print image pixel coordinates for given sky coordinates
skycoor	Convert between J2000, B1950, galactic, and ecliptic coordinates
sp2char	Replaces space in string with specified character (default=_)
subpix	Subtract a constant value(s) from specified pixel(s)
sumpix	Total pixel values in row, column, or specified area
wcshead	Print basic world coordinate system information for images
xy2sky	Print sky coordinates for given image pixel coordinates
	<program name> help lists possible arguments
	<program name> version lists version of program

3. Extending Keywords Beyond 8 Characters

In their 1981 paper, “FITS: a Flexible Image Transport System,” Wells, Greisen, and Harten, defined the standard rows for the components of a keyword line as shown in Table 2.

Table 2. FITS header spacing from original paper

COLUMN	111111111122222222223333333334									
NUMBERS	1234567890	1234567890	1234567890	1234567890	1234567890					
Card	1	SIMPLE	=						T	/ comment
Card	2	BITPIX	=						16	/ comment
Card	3	NAXIS	=						2	/ comment
Card	4	NAXIS1	=						190	/ comment
Card	5	NAXIS2	=						244	/ comment
Card	6	END								

3.1. ESO Solution

Over time, it has been hard to give such 8-character keywords an adequately descriptive name, so ESO adopted a new keyword, HIERARCH, which indicates that the real keyword name which follows is of arbitrary length and that the entire line no longer follows the original alignment as shown in Table 3

Table 3. ESO hierarch long FITS header keywords

```
HIERARCH CFA TEST LONGKEYWORD = 'This is a test'
$ gethead testbin.fits "hierarch cfa test longkeyword"
or
$ gethead testbin.fits "cfa test longkeyword"
This is a test
```

3.2. A Simpler Solution

Because the WCSTools FITS header reader already parses the line, and the HIERARCH option requires that the line be parsed using the “=” and “/” characters as field breaks anyway, why not simply allow arbitrary-length keywords as shown in Table 4.

4. New Features Coming to WCSTools

A single journal paper describing all of WCSTools will be written over the next year or so. Despite being in development for almost 25 years, this has been something of a side project, and there is quite a bit to be done before all of the necessary parts of the package are ready to undergo refereeing. Some of planned upgrades for the immediate future include remote access to the GAIA and most recent SDSS catalogs, local access to the UCAC5 and Atlas catalogs, improved WCS fitting algorithm, HEALPIX projection support, and Python wrapping of all of the tasks.

Table 4. Arbitrarily long FITS header keywords

```

OLD:
$ sethead testbin.fits longkeyword="This is a test"
$ gethead -e testbin.fits longkeyw
LONGKEYW= 'This is a test'
$ gethead testbin.fits longkeyw
This is a test
NEW:
$ sethead testbin.fits longkeyword="This is a test"
$ gethead -e testbin.fits longkeyword
LONGKEYWORD = 'This is a test'
$ gethead -e testbin.fits longkeyword
This is a test

```

References

- Calabretta, M. R., & Greisen, E. W. 2002, *A&A*, 395, 1077
- Greisen, E. W., & Cotton, W. 1994, *Classic AIPS World Coordinate Systems*. URL <http://tdc-www.harvard.edu/software/wcstools/wcstools.aips.html>
- Mink, D. J. 1996, in *Astronomical Data Analysis Software and Systems V*, edited by G. H. Jacoby, & J. Barnes, vol. 101 of *Astronomical Society of the Pacific Conference Series*, 96
- 1997, in *Astronomical Data Analysis Software and Systems VI*, edited by G. Hunt, & H. Payne, vol. 125 of *Astronomical Society of the Pacific Conference Series*, 249
- 1999, in *Astronomical Data Analysis Software and Systems VIII*, edited by D. M. Mehringer, R. L. Plante, & D. A. Roberts, vol. 172 of *Astronomical Society of the Pacific Conference Series*, 498
- 2011, *WCSTools: Image Astrometry Toolkit*. *Astrophysics Source Code Library*, 1109.015
- Wells, D. C., Greisen, E. W., & Harten, R. H. 1981, *A&AS*, 44, 363



From right to left: Jessica Mink, Anne Raugh, Alberto Accomazzi, Gus Muench, Peter Williams (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

ESAC Science Exploitation and Preservation Platform Reference Architecture

V. Navarro,¹ R. Álvarez,¹ F. Pérez-López,² C. Arviset,¹ J. Ventura-Traveset,¹
R. Prieto,¹ and A. Martín Furones³

¹*European Space Agency (ESA), Villanueva de la Cañada, Madrid, Spain*

²*RHEA for ESA, Villanueva de la Cañada, Madrid, Spain*

³*Politechnical University of Valencia, Spain*

Abstract. At ESA, active science missions like Gaia, Planck and XMM-Newton have developed precursor systems enabling the provision of advanced applications for the execution and instantiation of data analysis pipelines. Simultaneously, developments in missions like Euclid, as well as programmes like Galileo and Copernicus are tackling the creation of cyberinfrastructures capable of acquiring, processing, distributing and analysing massive amounts of data in an effective way. These initiatives have led to the implementation of solutions, commonly known as Thematic Exploitation Platforms.

ESAC Science Exploitation and Preservation Platform (SEPP) project drives the consolidation of past experiences and future needs into a reference framework to foster research through the provision of space science data, products and services. SEPP aims at integrating information and processing assets into a single environment to deliver advanced analysis and collaboration services.

This work presents SEPP's multi-mission reference architecture, which leverages on mainstream big data, cloud, virtualisation and container technologies to create a software as a service (SaaS) computing environment. This environment pivots around the paradigm shift characterised by the move of processing components to the data, rather than the move of data to the users.

1. Introduction

Big Data from Space refers to Space and Earth observation data collected by space-borne and ground-based sensors, as well as other space domains such as Satellite Navigation and Satellite Telecommunications. Whether for Space or Earth observation, they qualify being called “big data” given the sheer volume of sensed data.

The increase in the volume of information associated to Big Data challenges traditional scientific analysis methods. The initial download of products from data centre archives followed up by further processing at the different research institutions is experimenting a gradual paradigm shift where data centres do not only host the products but also the processing resources in charge of carrying out on-demand, remote analysis from the aforementioned research institutions.

Precursor activities like DAS-LT (Navarro 2015), Gaia AVI (Navarro et al. 2017), Planck AVI and XMM RISA (Ibarra et al. 2017) enable the provision of advanced col-

laboration systems for execution of data analysis processors that can be deployed in the archives, close to the data. Similar cyberinfrastructures are at the focus of missions like Euclid and programmes like Galileo or Copernicus.

Despite having similar needs and objectives, these cyberinfrastructure have been traditionally implemented in the form of vertical systems across different missions. This approach leads to ad-hoc or basic mechanisms to exchange information among them. At present, Thematic Exploitation Platforms have emerged as a system concept that encompasses functional and technical requirements commonly found in the implementation of these cyberinfrastructures.

The Science Exploitation and Preservation Platform (SEPP) consolidates requirements and experiences, integrating information and processing assets into a Virtual Research Environment to deliver advanced analysis and collaboration services.

SEPP’s multi-mission reference architecture leverages on mainstream big data, cloud, virtualisation and container technologies to create a software as a service (SaaS) computing environment. This environment pivots around the paradigm shift characterised by the move of processing components to the data, rather than the move of data to the users to fulfil a set of scenarios for data and ICT provision.

2. System Engineering Approach

SEPP’s system engineering approach runs several projects in parallel adhering to Agile Software Development principles (Navarro et al. 2019). The work follows an iterative, incremental life cycle, with the System Concept, Design and Implementation tasks executed in time-boxed sprints. Two overarching system engineering projects drive the strategic top-down definition of the solution. Alternatively, tactic, bottom-up activities are to be fed from on-going IT activities triggered by mission demands. The design of the SEPP must anticipate common needs in order to define an appropriate set of features that will support the development of future mission specific extensions on top of it. Therefore, bi-directional communication channels with present and future missions have been put in place to ensure SEPP alignment to mission needs. Use cases specified by these missions represent SEPP’s first requirements baseline. Among all contributing ESA missions, a higher level of involvement has been committed for the so called “early adopters.”

Planck	BepiColombo	Euclid
XMM	Galileo	Plato
JWST	Gaia	Archives

Figure 1. SEPP Missions

3. High Level Architecture

As a Software Platform, SEPP provides a semi-complete adaptable software infrastructure to be fully specified by the deployment context. Moreover SEPP acts as a coordination agent, gathering experiences from different projects that require solutions for similar challenges. This enables adoption and reuse of lessons learnt on common architectural patterns and technologies. SEPP's High Level Architecture is organised into several layers that resemble a typical N-tier architecture:

User Layer: this layer provides the HMIs for users and administrators to access all SEPP functionalities. The objective of this SEPP layer is to decouple presentation logic from business logic implemented by other layers. This layer allows smooth integration of HMI functionalities into a homogeneous look & feel through the provision of extension points.

Exploitation and Preservation Layer: this layer groups domains implementing generic and user specific analysis functionalities provided by SEPP. This layer groups domains implementing access to information and processing asset integrated in SEPP. The system will rely on executable modules and data at different levels of processing which will be natively stored on SEPP or federated to other systems. These two types of assets will be combined in the exploitation layer to deliver more complex services and products.

Business Logic Layer: this layer represents the area to be used for domain specific data processing, extensions and configurations. Business Logic Layer may be implemented as an extension inside SEPP platform or via the Interoperability Services. Support Layer: this layer groups domains implementing components that include libraries providing common features required by multiple domains across the whole system. These libraries are likely to require glue-code to adapt to specific component needs.

Infrastructure Layer: this layer provides basic support for the implementation of the preservation and support layers. It is based on COTS that can be reused "as-is" with an integration pattern mainly based on the configuration of a set of parameters to adapt the behaviour of the COTS to the specific needs.

4. SEPP Initial Services

SEPP puts the focus on the science community, to promote their contributions and involvement in the form of data and processing extensions with features such as:

- Interactive and collaborative data analysis (Jupyter, Octave, Zeppelin, R-Studio).
- Storage space in the platform for user's to bring their data and processing code close to the archives.
- Execution of user's custom pipelines in the platform.
- Interoperability and collaboration via VO protocols.
- Publication of science products and processors through a "Science App Store."

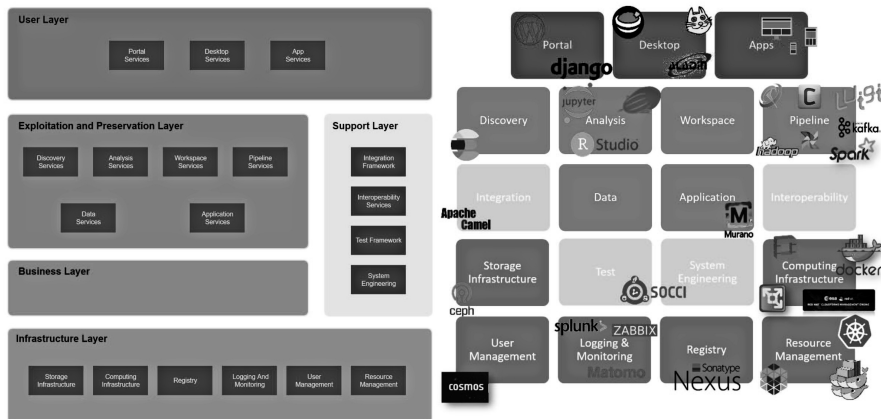


Figure 2. Reference architecture and technology stack

- Crowdsourcing pipelines for processing of massive, highly distributed datafeeds.
- Web based instantiation and access to legacy systems.

SEPP Workspaces represent the pivoting point providing seamless integration of analysis tools with data stored in user (persistent & volatile) or public (mission archives) areas. Workspace sharing features enable collaboration across research groups.

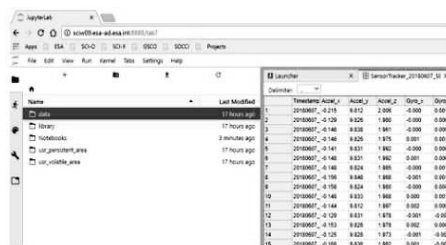


Figure 3. SEPP Workspace for JupyterLab

References

- Ibarra, A., Sarmiento, M., Colomo, E., Loiseau, N., Salgado, J., & Gabriel, C. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512, 389
- Navarro, V. 2015, in *Science Operations 2015: Science Data Management - An ESO/ESA Workshop*, 1
- Navarro, V., Vagg, D., O’Callaghan, D., McBreen, S., Hanlon, L., Hernandez, J., Salgado, J., & O’Mullane, W. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512, 169
- Navarro, V., et al. 2019, in *ADASS XXVI*, edited by M. Molinaro, K. Shortridge, & P. Pasian (San Francisco: ASP), vol. 521 of ASP Conf. Ser., 224

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Stellar Atmospheric Parameters from Full Spectrum Fitting of Intermediate- and High-resolution Spectra against PHOENIX/BT-Settl Synthetic Stellar Atmospheres

Evgenii Rubtsov^{1,2,*}, Igor Chilingarian^{1,3}, Svyatoslav Borisov^{1,2}, and Ivan Katkov^{4,1}

¹*Sternberg Astronomical Institute, M.V.Lomonosov Moscow State University, Universitetsky prospect 13, Moscow, 119234, Russia*

²*Faculty of physics, M.V.Lomonosov Moscow State University, 1 Vorobyovy Gory, Moscow, 119991, Russia*

³*Smithsonian Astrophysical Observatory, 60 Garden St., Cambridge, MA 02138, USA*

⁴*New York University Abu Dhabi, Saadiyat Marina District, Abu Dhabi, UAE*

**ev.rubtcov@physics.msu.ru*

Abstract. We present a new technique implemented in IDL for determination of the parameters of stellar atmospheres using PHOENIX and BT-Settl synthetic stellar spectra. The synthetic spectra provide good coverage in the T_{eff} , $\log(g)$, $[\text{Fe}/\text{H}]$, $[\alpha/\text{H}]$ parameter space over a wide wavelength range and allow fitting observed spectra of a vast majority of stars. Our procedure also determines radial velocities and stellar rotation and it takes into account flux calibration imperfections by fitting a polynomial continuum. Thanks to using pixel fitting we can exclude certain spectral features which are not present in the models such as emission lines (chromospheric emission in late-type stars or discs around Be stars). We perform a non-linear χ^2 minimization with the Levenberg-Marquardt method that is applied to the entire spectrum with the exception of areas with peculiarities: emission lines, model shortcomings (incompleteness of the spectral line lists used for the atmospheric model calculation). We compare the results of the analysis of optical spectra from ELODIE and INDO-US stellar libraries.

1. Introduction

The analysis of galaxy spectra has been historically performed using stellar population models which can be constructed in different ways. One can model stellar populations using synthetic stellar atmospheres which can be calculated for an arbitrary range in the T_{eff} , $\log(g)$, $[\text{Fe}/\text{H}]$ and $[\alpha/\text{H}]$ parameter space. The obvious drawback of this approach is that the list of spectral lines is never complete which means that stellar population model will inherit missing lines and will be discrepant with real galaxy spectra. On the other hand stellar populations modeled using observed stellar spectra match real galaxy spectra substantially better but observed spectra of stars used for modeling always cover a smaller region of the parameter space compared to synthetic spectra and are subject to imperfections in data reduction and calibration or a given star itself may be peculiar.

These problems can be partially solved by the use of semi-empirical models of stellar populations. These models are based on spectra from empirical stellar libraries complemented with synthetic spectra in the regions of the parameter space not well covered by empirical spectra. It is crucially important to accurately determine stellar atmospheric parameters of observed stars in order to correctly combine empirical and the synthetic grids of stellar spectra. Our work addresses the latter problem by using full spectrum fitting.

2. Pre-processing of Spectral Libraries

In this work we used PHOENIX (Husser et al. 2013) and BT-Settl (Allard et al. 2013) libraries of synthetic stellar atmospheres and INDO-US (Valdes et al. 2004), ELODIE (Prugniel et al. 2007), and UVES-POP (Bagnulo et al. 2003) empirical libraries of stellar spectra. For the parameter determination of stellar atmospheres we have prepared several grids of synthetic stellar spectra based on PHOENIX and BT-Settl libraries in order to cover a large range of parameters. We complemented the PHOENIX library at the high-temperature end ($T_{\text{eff}} > 15000$ K) by BT-Settl spectra, convolved them to the desired spectral resolution values and kept the two versions of each grid in vacuum and atmospheric wavelengths (Morton 2000). We generated a set of such synthetic grids for every value of the abundance of α -elements available in Phoenix ($[\alpha/\text{H}] = -0.4, -0.2, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2$ dex). As a result we have constructed grids of synthetic stellar spectra at spectral resolutions $R = 5000, 10000$, and 20000 in the wavelength range from 300 nm to 2.5 μm in a very wide range of atmospheric parameters ($T_{\text{eff}}: 2300 \dots 70000$ K, $\log(g): -0.5 \dots 6.5$, $[\text{Fe}/\text{H}]: -4.0 \dots 1.0$ dex, $[\alpha/\text{H}]: -0.4 \dots 1.2$ dex).

The preparation of empirical libraries of stellar spectra included post-processing of every individual spectrum. The INDO-US library ($R = 5000$) was fully re-assembled and re-calibrated in flux. We applied the telluric correction procedure developed for MagE data reduction pipeline, then ran a procedure for stitching non-intersecting spectral setups which used the model approximation, determined spectral line-spread function (LSF), corrected global imperfections of continua, and determined masks of poorly modeled spectral lines in several bins by T_{eff} . For the ELODIE spectra, we determined the LSF, corrected the continuum, and also calculated the line masks at $R = 10000$. The UVES-POP stellar spectra library required complete re-reduction of the original data using a new version of the UVES pipeline with adhoc correction for diffuse light and Echelle order stitching. We reduced its resolution to $R = 20000$. The entire process was briefly described in Borisov et al. (2018).

3. Description of the Spectrum Fitting Technique

The core of our method is a multidimensional interpolation procedure which produces a synthetic spectrum for any set of stellar atmospheric parameters and a non-linear minimization technique. At input, it requires a regular grid of spectra, for example, the one we described above in Section 2 which then undergoes the multi-dimensional spline interpolation and then is used in a non-linear χ^2 minimization using the Levenberg-Marquardt method (Markwardt 2009). During the minimization, in addition to the three parameters of stellar atmospheres (T_{eff} , $\log(g)$, $[\text{Fe}/\text{H}]$ for a fixed value of $[\alpha/\text{H}]$), the procedure also calculates a global radial velocity and the projection of a rotational

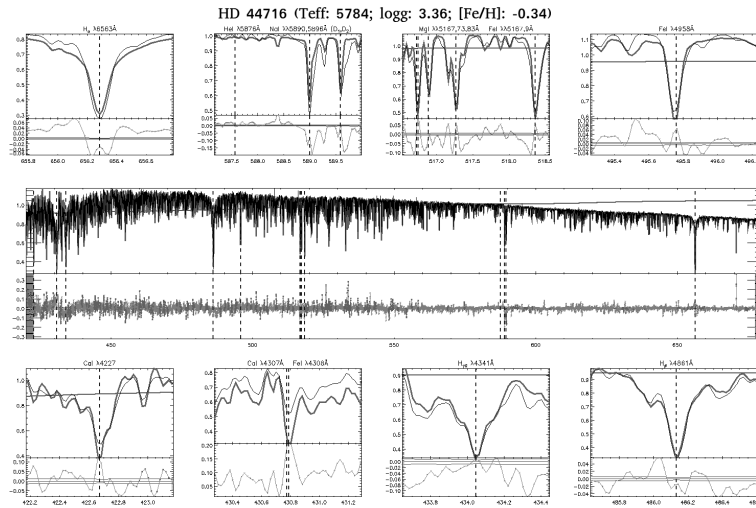


Figure 1. Application of our spectrum fitting technique to HD 44716 from the ELODIE library. Central panel shows the observed spectrum in black the best-fitting model in red and fitting residuals in green with masked regions shown in red. Smaller panels zoom in on the profiles of several spectral lines.

velocity of a star ($v \sin(i)$). Optionally we can account for residual wavelength calibration errors using a low-order polynomial function. Because the global continuum of an empirical spectrum is subject to flux calibration errors we correct it with a low-order Legendre polynomial in order to bring a spectrum closer to the model. The observed spectra may possess some features that are not well modeled in synthetic spectra, for example, some spectral lines coming from chromospheric emission or interstellar absorption, also data reduction imperfections like cosmic ray hits, etc. In order to ensure that such artifacts are not affecting the minimization we construct masks that exclude these spectral regions. At the end our procedure performs the χ^2 -minimization over the entire spectrum excluding the masked regions with pixel weights inversely proportional to flux uncertainties. An example of our technique applied to a stellar spectrum is shown in Fig. 1. The code is written in *IDL* using *MPFIT* and *LAPACK* packages. It allows us to fix and/or tune all components of the fitting procedure. We plan to re-write the code into *Python* in the near future and make it publicly available.

4. Comparison of the Fitting Results for Spectra from Different Stellar Libraries

We tested our approach using Monte Carlo simulations which showed excellent convergence and independence of the solution from the initial guess. We then applied our technique to observed stellar spectra from the stellar libraries ELODIE (1959 spectra reduced to $R = 10000$), INDO-US (1273 spectra, $R = 5000$) and UVES-POP (408 spectra, reduced to $R = 20000$) at wavelengths longer than 420 nm. To determine the parameters of stellar atmospheres the spectra of each library were corrected for a global continuum. We used pixel masks created for each library as described earlier.

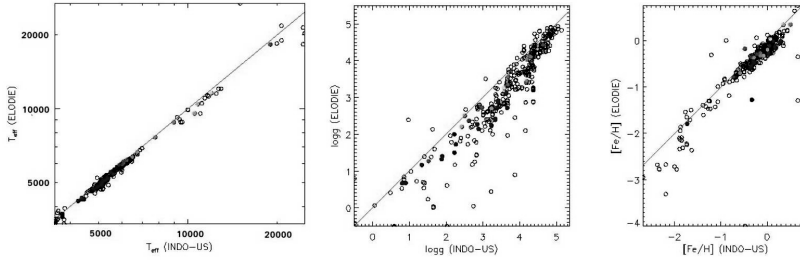


Figure 2. Comparison of the fitting results between ELODIE and INDO-US stellar libraries. The panel left to right shows a comparison of effective temperature, surface gravity, and metallicity. Empty circles show all cross-matched stars between ELODIE and INDO-US (408 spectra). Colored circles show the subset of stars for which CDS Simbad reports the object type “*” (35 spectra).

We compare the derived atmospheric parameters for the overlapping sub-sample of stars in the ELODIE and INDO-US stellar libraries (Fig. 2). In general, there is a good agreement between the fitting results for effective temperature and metallicity for the full sub-sample. There is a slight systematic difference in $\log g$ values which might be the result of using different fitting wavelength ranges for the two stellar libraries, because the quality of synthetic stellar spectra changes across wavelengths. Some stars can also be discrepant because of spectral binarity and/or variability.

Our method allows us to determine the stellar atmospheric parameters for the stellar spectra of all spectral classes and metallicities within the framework of synthetic grids. For each spectrum individual fine tuning of the fitting parameters is possible. Accounting for the polynomial continuum allows one to account for flux calibration imperfections that distort the global continuum shape. The construction of pixel masks eliminates features badly modeled in synthetic atmospheres and data reduction artifacts in the observed spectra. All this together allows us to determine parameters of stellar atmospheres with the accuracy sufficient for stellar population modeling.

Acknowledgments. This project is supported by the Russian Science Foundation Grant 17-72-20119. ER is grateful to the ADASS-XXVIII organizing committee for providing financial aid to support his attendance of the conference.

References

- Allard, F., Homeier, D., Freytag, B., Schaffenberger, W., & Rajpurohit, A. S. 2013, *Mem. S.A.It.*, 24, 128. [arXiv:1302.6559](#)
- Bagnulo, S., Jehin, E., Ledoux, R., C. and Cabanac, Melo, C., Gilmozzi, R., & ESO Paranal Science Operations Team 2003, *The Messenger*, 114, 10
- Borisov, S., Chilingarian, I., Rubtsov, E., Ledoux, C., Melo, C., & Grishin, K. 2018, [arXiv:1802.03570](#)
- Husser, T. O., Wende-von Berg, S., Dreizler, S., Homeier, D., Reiners, A., Barman, T., & Hauschildt, P. H. 2013, [arXiv:1303.5632](#)
- Markwardt, C. B. 2009, *ASPCs*, 411, 251. [arXiv:0902.2850](#)
- Morton, D. C. 2000, *ApJS*, 130, 2
- Prugniel, P., Soubiran, C., Koleva, M., & Le Borgne, D. 2007, [arXiv:astro-ph/0703658](#)
- Valdes, F., Gupta, R., Rose, J. A., Singh, H. P., & Bell, D. J. 2004, *ApJS*, 152, 251

Session VI

Quality Assurance of Science Data

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

The BagIt Packaging Standard for Interoperability and Preservation

Raymond Plante,¹ Gretchen Greene,² and Robert Hanisch³

National Institute of Standards and Technology, Gaithersburg, MD, USA;

¹*raymond.plante@nist.gov*, ²*gretchen.greene@nist.gov*,

³*robert.hanisch@nist.gov*

Abstract. BagIt is a simple, self-describing format for packaging related data files together which is gaining traction across many research data systems and research fields. One of its great advantages is in how it allows a community to define and document a profile on that standard—that is, additional requirements on top of the BagIt standard that speak to the needs of that community. In this paper, we summarize the key features of the standard, highlight some important profiles that have been defined by communities, and discuss how this standard is being used as part of the NIST Public Data Repository as a preservation format. We will compare and contrast the use of BagIt generally for enabling interoperability (e.g., for transferring data between two systems) and its use for preservation. We will then give an overview of the NIST BagIt profile for preservation as well as introduce a general-purpose MultiBag profile which addresses issues of evolving data and scaling to large datasets.

1. Introduction: the Data Publication Backdrop

Across all disciplines of research, there is a growing recognition of the importance of data in the scientific process, the work necessary to make data science-ready, and the great value data provides to the greater scientific endeavor when it is shared. For that value to be realized, however, care is required by a chain of data stewards—authors, curators, and publishers—to ensure that the data is available via open formats, that they are sufficiently annotated with metadata that make them understandable to others, and that the metadata are indexed in intra-community discovery systems so that others can find it. Recently, there has emerged a unifying perspective on how to realize the value of shared data which is known as the FAIR data principles. These principles spell out what it means to make data Findable, Accessible, Interoperable, and Reusable (Wilkinson et al. 2016). Data repositories and their curators play an important role in making data FAIR.

Data creators—or authors, we can call them—play a key role, too, and with that role comes real work and effort. Consequently, we cannot expect to reap the benefits of FAIR data without giving proper credit and recognition to the authors that make their data FAIR. Thus, there is now a robust conversation happening across disciplines about how we communicate that credit in a manner that makes a positive impact on authors' careers: the emerging currency of that credit is the data citation—a recognition of a scientific output equal to that of the traditional literature citation (Data Citation Synthesis Group 2014; Meunch & D'Abrusco 2019). In other words, scientists must be

afforded another channel for communicating scientific results that can be just as potent as a literature publication: that is the *data publication*.

Again, data repositories play a big role in making the data they *publish* FAIR, and many have taken up responsibility of socializing the concept of the data publication in the way they present data. As an example, one can consider Muench et al. (2015), a presentation of infrared images of the Trapezium cluster, provided via the CfA Dataverse (Crosas 2011). The landing page for this data publication bears many of the hallmarks of a traditional literature publication: it has a title, a list of authors, the equivalent of an abstract, and access to its contents—in this case, the data files. Very importantly, the recommended way to cite this data publication is featured prominently. We also have access to all the publication's metadata, both at the dataset and file levels. As another example, we can consider Getman et al. (2017) from the Zenodo repository¹. Its landing page has all the same hallmarks, but this one also displays a preview of the data in the form of a catalog plot overlaying an image. We can see that this preview is presented via an in-lined tool with rudimentary zooming ability. From these examples, we can see that a repository can be more than a place to dump and download data, but actually another channel for communicating scientific results.


The National Institute of Standards and Technology (NIST) has recently brought on-line its own institutional repository, the NIST Public Data Repository (data.nist.gov). From its inception, we have viewed this repository as a publication platform in the tradition of our institutional journal, *The Journal of Research of NIST*—that is, as a channel for communicating results from NIST researchers. More than that, we have endeavored to make the repository an exemplar for data publishing for disciplines that are perhaps less practiced at data sharing than astronomy. Like the aforementioned repositories, we have adopted a publication style for landing pages that is meant to encourage re-use and citation (see Fig. 1). Forthcoming features like a data cart, data previews, and embedded tools will make it easier to browse and download the data.

There are two features of our presentation worth noting because they affect the preservation of the underlying data. First, we allow authors to organize their data files into hierarchical collections; this is intended to make it easier to browse a publication made of many files. The most important or most summarizing data files can be put at the top of the hierarchy, and detailed data files can be segregated into subcollections. Second is to note that metadata is everywhere, both at the publication and file level (like Dataverse). The landing pages, of course, are generated on-the-fly from that metadata which ultimately come from the authors. In our next major release of the repository, we will be embedding tools that allow authors to customize their landing pages. They will be able to click on the references and add additional entries, they will be able to add ORCIDs² to the author list, they will be able to refine or expand on their abstract, description, and keywords. In reality, they will be editing the underlying metadata, but from their perspective, they are customizing the presentation to improve the message of their publication. If we can make this easy to do, then we want to let them do this often, even after the initial public release.

Through the NIST Public Data Repository, we are trying to get scientists to *care* about how their data is presented. We want to give them tools to take control of that

¹<https://zenodo.org/>

²<https://orcid.org>



Public
Data Repository

1.1.0-beta

About | Search

Data Publication

Experimental test of the intrinsic dimensionality of Hounsfield unit measurements: the CT data

Z. H. Levine, A. R. Peskin, A. Holmgren, E. Garboczi

Contact: Zachary Levine...

Identifier: doi:10.18434/M3M956

Version: 1.1... Last modified: 2018-05-18

Visit Home Page

Go To ..

- Description
- Data Access

Record Details

- View Metadata
- Export JSON

Use

- Citation
- Fair Use Statement

Find

- Similar Resources
- Resources by Authors

We present the data supporting "Experimental test of the intrinsic dimensionality of Hounsfield unit measurements" (In preparation). In this study, we passed 34 different substances in separate vials through a computed tomography (CT) scanner at 4 different voltages. At each voltage, we obtained 1824 images (in DICOM format) depicting a sequence of slices through the vials. All 7296 images are provided here. In addition, we provide a table of the substances, their masses, and their positions in the sequence. This dataset deprecates the earlier release of this data (ark:/88434/mds019bfm9). The image and substance table data are exactly the same; however, the image data has been re-arranged to make browsing and downloading more convenient.

Subject Keywords: x-ray computed tomography, medical phantom, Hounsfield unit, volume, shape

Data Access

These data are public.

Files

Click on the filerow in the table below to view more details.

Total No. files: 475

Name	Media Type	Size	Download
<div> Compounds </div> <div> 080kV </div> <div> 100kV </div>			
slices0001-0100.zip	application/zip	13.3 MB	Download
slices0101-0200.zip	application/zip	15.1 MB	Download

Figure 1. A sample landing page for a data publication in the NIST Public Data Repository (Levine et al. 2018). This view is generated on-the-fly based on publication metadata. Note that the files can be organized into a hierarchy.

presentation, and we want them to use their data publication as another chronicle in the story of their scientific findings.

2. Introduction: a Preservation Primer

A typical data repository has as part of its ingest workflow a step in which a copy of the prepared data collection is saved to long-term storage (perhaps as multiple, distributed copies). The form of that data is often not exactly how it is stored in the active archive serving users; rather, the form is optimized for preservation goals. First, this tucked-away copy exists in case there is a need to restore data into the active archive from its original form: a single file may get corrupted and need to be restored from the safer long-term storage. Alternatively, an entire repository may need to be restored from scratch in the case of a major hardware failure. In either case, storing checksums for every file is critical for detecting corruption.

Another important preservation goal is about ensuring that the data remains accessible, readable, and understandable long into the future (where “long” is measured in at least decades). Part of a robust preservation plan looks at how the data can survive changes in hardware and operating systems. Such a plan must also consider the data formats used. Traditional wisdom would encourage the use of the well-documented formats; that way, new software can be engineered to read the data when the original software that wrote the data no longer runs. Further, one would like a preservation format that is self-documenting; the next generation of curators then have a chance of *understanding* the data long after the data creators have retired or are otherwise inaccessible.

As we can see from the previous introductory section, data publications are more than just a pile of FITS files. They include metadata, documentation, previews, and other ancillary data. A format that can store all of those things together stands a better chance of preserving the meaning and message of that data.

3. The BagIt Format

BagIt is a packaging format for transmitting a digital collection. It was developed by archivists at the California Digital Library, the US Library of Congress (LOC), and Stanford University (Kunze et al. 2018). A key use case for BagIt is as a format for delivering a data collection to an archive to be ingested as a coherent whole. The BagIt creators have used the IETF document process to develop it as a specification; now in its 17th revision, IETF Request For Comment (RFC) 8493 recently reached version 1.0 status as an informational document. The LOC maintains supporting software libraries written in Java and Python³; both include a validator and a command-line tool. As the specification has matured, the format has seen broader adoption as a format for transferring collections between platforms. In particular, some research repositories offer BagIt as an export format for data collections. At NIST, we have adopted BagIt as a *preservation format* due to some of its particular advantages.

A collection compliant with the BagIt specification is referred to as a “bag”, and a bag is fundamentally a directory on a filesystem with some prescribed contents. In particular, it must include,

- a subdirectory called **data** that contains the collection “payload”—i.e., the primary files constitute the collection,
- a small file called **bagit.txt** which identifies the directory as a bag,
- a file called **bag-info.txt** which contains minimal, machine-readable (but human-oriented) metadata,
- one or more files named **manifest-*alg*.txt** where *alg* identifies a checksum hashing algorithm (e.g., **md5** or **sha256**); such a file lists the names of the files under the **data** directory along with their checksum values.

³<https://github.com/LibraryOfCongress/bagit-java>; and
<https://github.com/LibraryOfCongress/bagit-python>

Beyond these simple requirements, a bag provides a lot of flexibility. The files under the **data** directory can be organized into an arbitrary hierarchy of subdirectories. The root directory of the bag can contain any other files or subdirectories for arbitrary content; this is the typical way to include site-specific metadata about the collection. Finally, the bag directory may be serialized and compressed into a single file in any manner (such as into **zip**, **tar.gz**, or **7z**) for transmission or storage.

Another feature of BagIt bags that is interesting for very large collection is that it may contain an optional file called **fetch.txt**. Each line in the file includes the file path to a payload file (relative to the **data** directory), the file's size in bytes, and a URL from which the file can be retrieved. Any file listed in the **fetch.txt** is not required to appear under the **data** directory. It is thus possible to transfer a small bag that contains a **fetch.txt**; the receiver can then decide whether to retrieve all or any of the listed files to reconstitute the original collection. This is a useful feature when using BagIt to transfer collections, but it's not so good for the preservation context, as we will see.

3.1. BagIt Profiles

Because the BagIt specification allows any other additional files outside of payload (as well as additional non-standard metadata in the **bag-info.txt** file), some communities have taken to defining their own customizations referred to as a *profile*. A profile is a set of additional rules on top of the BagIt specification regarding a bag's contents. In fact, there is a project in GitHub called **bagit-profiles**⁴ that defines a JSON-formatted description of the requirements of a BagIt profile. It also provides a validator that, given a profile description, can determine if a particular bag is compliant with the profile.

One example of a BagIt profile has been defined by the DataONE project⁵. DataONE is a data grid of distributed repositories, services, and tools sharing and using Earth and environmental data. In their profile, a bag includes an ORE file compliant with the OAI Object Reuse and Exchange standard (Lagoze et al. 2008). An ORE document describes data aggregations, particularly relationships between the files, using RDF (Klyne & Carroll 2004). This makes it possible to describe roles and purposes of the various files.

Another profile of note is one recently defined by the Research Data Alliance's Repository Interoperability Working Group (RDA Interoperability WG 2018). The aim of this profile is to provide greater interoperability between repositories and data preparation systems by requiring some minimal metadata for understanding the contents. In this profile, the bag's root directory must contain a subdirectory called **metadata** which in turn contains a file called **datacite.xml**. This XML file uses the DataCite schema (DataCite Metadata Working Group 2017) to describe the collection; this is the schema used for the metadata behind the DataCite digital object identifiers (DOIs). As collections that might typically be packaged according to this profile would either already have an assigned DataCite DOI or soon would be after ingest into a repository, providing this metadata would not be difficult.

It is worth noting that it is typically possible for a bag to be compliant with more than one profile simultaneously.

⁴<https://github.com/bagit-profiles/bagit-profiles>

⁵<https://releases.dataone.org/online/api-documentation-v2.0/design/DataPackage.html>

3.2. BagIt for Preservation at NIST

As we can see, the BagIt format provides a number of attractive features as a preservation format. All of the data can be stored together in a single package and in their original hierarchy as provided by the authors. We can also include full metadata descriptions in our local schema and format or even in multiple standard formats. We can also include any ancillary data like figures or other preview visualizations. We can serialize and compress our bags for long-term storage. (This is particularly useful to NIST as we are currently using Amazon S3 storage⁶ which does not have support for hierarchical filesystems.) It meets our key preservation requirements: we have file checksums to detect file corruption, bags can be made fully self-contained with no external dependencies, and, through the metadata we include, we can make them self-describing.

There are some disadvantages to using BagIt as a preservation format. The first comes with supporting very large collections (in terms of numbers of bytes). That is, it is not particularly convenient to store very large, serialized bags; in particular, restoring a single file from large serialized bag can be inefficient if it requires copying and decompressing the entire bag first. Using the **fetch.txt** file solution as a way of keeping bags small is not appropriate for preservation: not only are we no longer storing files together, we should not rely, decades down the road, on the existence of the original HTTP service to serve us the **fetch.txt** files.

The second challenge comes with handling versioning. As we discussed in the introduction, we want to encourage authors to improve the metadata for their data publications often, even after the initial release. This means we need to version the publication. We don't want to make a second copy of an entire bag when the only thing that has changed is a single metadata field.

Fortunately, the flexibility of the BagIt format and the practice of defining profiles open a door for addressing these disadvantages. Because we can describe the problem in general terms that need not be peculiar to the NIST repository, it makes sense to try to solve the problem in a general and open way. To that aim, we introduce what we are calling the Multibag BagIt Profile.

4. The Multibag Profile

The basic concept behind the Multibag Profile is that a (large) bag is split into several smaller bags, where each component bag is a BagIt-compliant bag on its own. The core BagIt standard has partial support for this—that is, one can identify a bag as being part of a group of related bags; the Multibag Profile goes further by defining additional metadata and rules for reconstituting the complete bag from its component bags. It also allows for efficient lookup of the location of individual files amongst the component bags. Finally, it provides a mechanism for future creation of “errata” bags that capture only changes to the collection (new or updated files); this allows for low-cost, non-destructive updates to collections previously archived. The profile is fully documented

⁶This commercial cloud service is identified here to adequately describe a constraint of the archiving system. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the service identified is necessarily the best available for the purpose.

in our GitHub repository, **multibag-py**⁷. The **docs** directory includes the latest version of the profile specification, but also, as the name suggests, it contains a reference library implemented in Python for creating, accessing, and reconstituting Multibag aggregations.

A Multibag aggregation is set of one or more bags that, when properly combined, represent a single coherent data collection. That set of bags includes one *head bag* and zero or more additional *member bags*. The head bag represents the entry point into the aggregated collection; consequently, it has additional requirements on its contents:

- The **bag-info.txt** file contains some additional metadata that help identify it as a head bag.
- The head bag includes a file called **multibag/member-bags.tsv**: this file lists the names of the other bags that make up the aggregation. (Optional URLs for those bags may also be included.)
- Also in the **multibag** subdirectory is a file called **file-lookup.tsv**: this lists the payload files (and optionally other ancillary files) that make up the aggregated bag, each mapped to the name of the member bag in which it is located.

The order of the member bags listed in **multibag/member-bags.tsv** is significant: it represents the order in which the bags should be unpacked into a common directory in order to reconstitute the complete bag. As each bag is unpacked, its files may override previously unpacked files.

The mechanism for creating an update to the aggregation takes advantage of this recipe for reconstituting an aggregated bag. An update is accomplished by creating a new head bag and zero or more additional member bags. The new bags only contain those files that are new or have changed. These changes can be in either payload files or metadata files. In the new head bag, the **member-bags.tsv** file can (and usually does) list member files that were part of the previous version of the collection. Further, the **bag-info.txt** file in the new head bag identifies the version of the revised aggregation. It can also make references to the head bags of the previous versions it deprecates. Finally, if any files should be considered as having been removed from the collection in the new version, the new head bag can contain the optional **deleted.txt** file that lists those files.

It's important to note that none of the bags created for a previous version are changed in any way. In fact, access to the member bags is not even necessary to create the update; access to just the previous head bag is sufficient. This means the update is *non-destructive*. Consequently, a revised multibag aggregation can actually contain multiple head bags, one for each of the available versions. Any version of the collection can be reconstituted by first selecting the appropriate head bag (see fig. 2).

We have already started leveraging this profile in the NIST Repository; nevertheless, there are features we envision adding in the future. First, the specification does not yet describe how to deal with an individual file that is inconveniently large; we plan to add rules for splitting up large files across member bags. Second, we would like to better leverage persistent identifiers (PIDs, such as DOIs) to identify the component bags in an aggregation. We mentioned that a generic URL locating a file may not reliably

⁷<https://github.com/usnistgov/multibag-py>

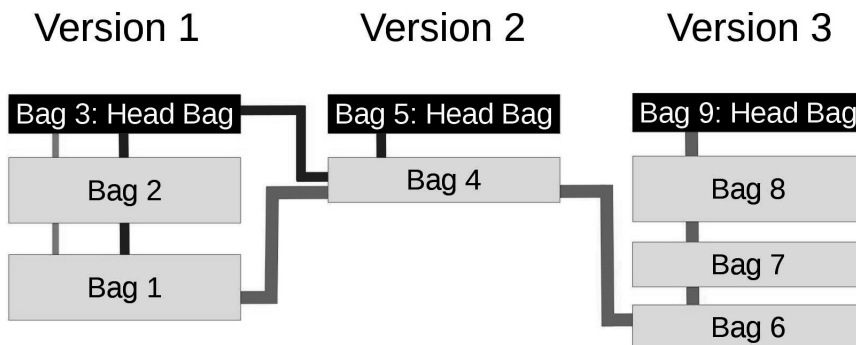


Figure 2. An illustration of non-destructive collection updates. The boxes represent bags that contain data from the collection; each colored line (of different thickness) connects the bags that are part of the same version of the bag aggregation. The head bag for a bag aggregation contains a **member-bags-tsv** file that lists what other bags are part of the aggregation. Version 1 of the aggregation (thin green line) includes Bags 1-3. Version 2 (medium blue line) includes the original Bags 1-3 plus Bags 4 and 5. Version 3 (thick red line) includes Bags 6-9 but only Bags 1 and 4 from the previous versions.

exist decades in the future; however, we can more likely expect that an updated PID resolver service that can point to a component bag *will* exist.

5. The NIST Preservation Profile

There are other preservation requirements that we need to address in our preservation format beyond handling large collections and updates to collections. Some of the requirements address needs and features that are peculiar to our repository. For this reason, we are also defining a NIST Preservation BagIt Profile. This profile documents the organization of additional information and data created by our repository as part of the data ingest process, some of which will be critical to restoring in case of system failure, and some which will help future NIST curators understand the contents of the bags.

The key feature of the NIST profile is the inclusion of **metadata** subdirectory. This directory includes:

- **pod.json** – a metadata description of the collection encoded in JSON-LD (Sporny et al. 2014) according to the Project Open Data Dataset schema⁸, a schema that is used across US government agencies.
- **nerdm.json** – a richer description of the collection, also in JSON-LD, using our internal repository schema (the NIST Extensible Resource Model, or NERDm).

⁸<https://project-open-data.cio.gov/v1.1/schema/>

- **datacite.xml** – a DataCite-compliant metadata description derived from the **nerdm.json** record. This makes our bags compatible with the RDA-recommended profile.
- **ore.json** – a resource map compliant with the OAI Object Reuse and Exchange (ORE) standard which encodes RDF statements about relationships between files in the bag (similar to what is done in the DataONE project).

The **metadata** directory also includes a directory hierarchy that mirrors the hierarchy below the **data** directory except that where the latter stores a data file, the metadata tree has a directory named after the data file. In that directory, we store metadata (in NERDm format) and other ancillary data that are specific to that data file. Ancillary data can include, in particular, a preview or visualization of the data in the data file.

Finally, our NIST profile includes a few extra files in the top directory:

- **preservation.log** – the log file from the ingest process that created the bag.
- **premis.xml** – preservation metadata compliant with the PREMIS standard format (PREMIS Editorial Committee 2015).
- **about.txt** – a plain-text, human-readable summary of the collection; this represents a “last resort” communication to the future archivist or other consumer who is otherwise unable to understand any of the other standard descriptions.

As of this writing, some of the features described in this section are still in development, and our profile is still evolving. Nevertheless, our goal is to produce a definitive, well-documented preservation format that can help ensure access to NIST data products long into the future.

6. Conclusion

We propose that the BagIt packaging format is very well-suited as a data preservation format in its ability to capture a safe, comprehensive, self-describing snapshot of a data publication; however, on its own, the format does have some short-comings for handling large and evolving collections. To close this gap, we have developed the Multibag BagIt Profile that provides a mechanism for dividing a collection cross multiple bags as well as a recipe for combining those bags back together to reconstitute the collection. The Multibag Profile, then, constitutes one component of the NIST Preservation Profile which addresses all of our requirements for preserving data publications in the NIST Public Data Repository.

A big part of the motivation behind the development of the NIST Repository is the desire to promote and encourage the practice of publishing data as a first-class artifact of scientific output. NIST has been in the business of publishing reference data for decades; the now emerging standards and infrastructure for finding, using, and citing data bring greater visibility to NIST data products. To ensure that proper credit flows back to the creators of these products, we must now re-examine all aspects of the data lifecycle. This includes data preservation, and so central to our requirements is the need to preserve the scientific message behind a data publication and ensure its usefulness well into the future.

References

- Crosas, M. 2011, D-Lib Magazine, Volume 17. URL <http://www.dlib.org/dlib/january11/crosas/01crosas.html>
- Data Citation Synthesis Group 2014, Joint declaration of data citation principles. Martone, M. (ed.) (San Diego: FORCE11). URL <https://doi.org/10.25490/a97f-egy>
- DataCite Metadata Working Group 2017, Datacite metadata schema documentation for the publication and citation of research. URL <https://doi.org/10.5438/0014>
- Getman, K., Broos, P., Kuhn, M., Feigelson, E., Richert, A., Ota, Y., Bate, M., & Garmire, G. 2017, Star Formation In Nearby Clouds (SFINC): X-ray And Infrared Source Catalogs And Membership. SPCM Atlas Dataset. URL <https://doi.org/10.5281/zenodo.231216>
- Klyne, G., & Carroll, J. J. 2004, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 10 Feb. 2004. URL <http://www.w3.org/TR/rdf-concepts/>
- Kunze, J., Littman, J., Madden, E., Scancella, J., & Adams, C. 2018, The BagIt File Packaging Format (V1.0), IETF RFC 8493. URL <https://tools.ietf.org/html/rfc8493>
- Lagoze, C., de Sompel, H. V., Johnston, P., Nelson, M., Sanderson, R., & Warner, S. 2008, ORE Specification - Abstract Data Model, OAI Specification. URL <https://www.openarchives.org/ore/1.0/datamodel>
- Levine, Z. H., Peskin, A. R., Holmgren, A., & Garboczi, E. 2018, Experimental test of the intrinsic dimensionality of Hounsfield unit measurements: the CT data. URL <https://doi.org/10.18434/M3M956>
- Meunch, A., & D'Abrusco, R. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 709
- Muench, A., Alves, J., Lada, C., & Lada, E. 2015, Replication data for: Deep 3.8 micron observations of the trapezium cluster. URL <https://doi.org/10.7910/DVN/28977>
- PREMIS Editorial Committee 2015, PREMIS Data Dictionary for Preservation Metadata, Version 3.0. URL <https://www.loc.gov/standards/premis/v3/>
- RDA Interoperability WG 2018, Research Data Repository Interoperability WG Final Recommendations. URL <https://doi.org/10.15497/RDA00025>
- Sporny, M., Longley, D., Kellogg, G., Lanthaler, M., & Linström, N. 2014, JSON-LD 1.0: a JSON-based Serialization for Linked Data, W3C Recommendation, 16 Jan. 2014. URL <https://www.w3.org/TR/json-ld/>
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 'T Hoen, P. A. C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., & Mons, B. 2016, Scientific Data, 3, 160018

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Adding Science Validation to the JWST Calibration Pipeline

Rosa I. Diaz¹ and M. Macarena Garcia Marin²

¹*STScI, 3700 San Martin Dr, Baltimore, MD 21218; rdiaz@stsci.edu*

²*European Space Agency, 3700 San Martin Dr, Baltimore, MD 21218*

Abstract. The JWST Calibration Pipeline will provide with the best calibration for all JWST instruments, observing modes, and a wide range of science cases. This software, organized into three main stages, needs careful scientific validation and verification. These are necessary to determine consistency and quality of the data produced by the calibration pipeline. With this goal in mind, the scientist at STScI have supported validation testing for most of the major builds.

1. Introduction

The Mission and Engineering and Science Analysis (MESA) branch embarked on the task to support the scientific validation of the calibration pipeline. This has now been a multi-year effort starting in 2016 and continuing even after launch. Given that the JWST Calibration Pipeline singlehanded performs calibration of all JWST data, coordinating the scientific validation with all the instrument teams is in particular important.

Our experience with the Hubble Space Telescope made us realize the level of effort needed to consistently and reliably validate the calibration pipeline after each build. This motivated us to think about ways to streamline the process to validate the JWST Calibration Pipeline without sacrificing the quality and depth of the testing. For this, we first defined a set of low-level unit tests to verify it produces the expected results. In addition to that, the instrument teams started to look for a more in-depth scientific validation for the wide range of science cases that will be observed by JWST. This later required a different strategy; one that validates the accuracy of the data and provides with reliable metrics for all science cases.

Although we already performed validation testing for four of the builds, we faced limitations on the depth of testing that could be done due to the state of development of the software and the availability of accurate test data. We also had to reconcile a diversity of ideas coming from a large group of scientists from different teams. We dealt with always changing resources and conflicting schedules. The end result is now a more mature plan that integrates the science validation with the build process. In here we discuss the steps we are taking to complete this effort. We also discuss the challenges we faced to make this possible and how this work will help us better support the development of the JWST Calibration Pipeline after launch.

2. The Calibration Pipeline

The JWST Calibration Pipeline is a Python software suite that automatically process the data taken by the JWST instruments: MIRI, NIRCam NIRISS, and NIRSpec. It supports all JWST observing modes and it produces fully calibrated products for individual exposures as well as high level products. The later are products that result from the combination of images and spectra exposures taken at different dither positions, in a mosaic, or even from different detectors within a single instrument. The JWST Calibration Pipeline is divided into tree different stages:

- Stage 1 processes raw detector data and produces uncalibrated slope images for all exposures
- Stage 2 calibrates the individual slope images according to the type of data: Imaging or spectroscopy. It produces calibrated slope images for all integrations and exposures.
- Stage 3, combines the slope images or spectra into a single product. It is further subdivided by the type of data.

This subdivision is further complicated by a diversity of steps done in each of these stages. For the Stage 1 module we have 13 different steps which are applied, or not, depending on the type of detector or type of observation. The usage of a given step depends also on the type of instrument (Near IR and Mid IR) and in some cases the type of observation (e.g., TSO, Darks). Stage 2, process imaging and spectroscopic data, and thus differences between the various modules are more evident. The steps, however, also depend on the type of detector and observation. Finally, for Stage 3 we have fewer steps per module but in this case we have a module per observation type (Coronagraphy, Imaging, Aperture Mask Interferometry, Time Series Observations, and Spectroscopy) (STScI 2016). As we can see, science validation is not a simple task.

3. Why is it Important to Involve the Science Staff?

Validating the calibration pipeline not only requires making sure the steps run. When the goal is to ensure mission requirements are met, we also need to, for each step:

1. assess that the Calibration Pipeline has full coverage for all the data that JWST will take,
2. ensure the reference calibration data is compatible with the calibration software,
3. verify the optional parameters for some of the steps are consistent with the defaults.

Also, we recognize that there are different ways to implement an algorithm, and that it is also subject to interpretation of the requirements or the clarity of the documentation. It is therefore important to determine whether the calibration software correctly implements the algorithms and the different options these offer. Scientists also need to go even further and determine how well the selected algorithms work for the different types of data JWST will take. This includes validation of the edge cases, the selected default values and thresholds, and the reference data used.

Initially, it was envisioned that the Data Management System (DMS) Integration & Testing team would do the scientific validation of the calibration pipeline together with the validation of the full DMS system. This, however, quickly became a challenge because of the conflicts of schedules as well as the different scope and goals of these two groups. Engineers might require verification of the full set of instruments and observation modes and might be more concerned about design, performance, optimization, and maintainability. Scientists, on the other hand, care about determining the accuracy of the calibration products for a large dynamic range of science cases, validating that the selected algorithms support exploration of a broad set of science parameters, assess the impact of each step has on the overall quality of the products, and even validate that the selected algorithms are adequate. Also, since the JWST Calibration Pipeline covers all instrument modes, the science validation for some of the steps can be done with data from a single instrument and for a subset of the observing modes.

The only way to achieve the goals of these two groups is to divide the work: I&T performs full JWST Pipeline verification and Instrument Science Teams perform science validation. This turned out to help speed up the process to fix bugs and to improve on the algorithms implementation. While at the beginning we were validating the calibration pipeline only after each build, after the second round, instrument teams started testing the steps as they became available. Instrument teams were also able to work as resources became available and to use cryo-test data as soon as it became available. It also allowed the science team to subdivide the scope of tests even more in order to have results faster; waiting until a full scale and complete validation could be done requires waiting until after we are inflight in order to get the science data needed.

The first type of testing we want to perform is to validate that the calibration pipeline code does what the science teams expect to see as defined by the algorithms and that these meet the science requirements defined at the start of the mission; we call this validation part 1. Once this is done, we can move to validation part 2; this is, to perform tests that allows the science teams to determine the accuracy and quality of the calibration products, to what extent the selected algorithms meet the error budget, and how these vary for different types of data or science cases. Some of these tests also require in-flight data. The science teams also want to repeat some of the same tests before the release of a new build of the Calibration Pipeline. This requires the allocation of a considerable number of resources to this effort as well to perform these tests in a short period of time, by all the instrument teams, and with the same testing standards.

4. Science Validation Plan

In order to simplify the process of scientific validation testing, we have to carefully define validation tests that can be performed automatically. Currently we have a list of about 426 tests for the calibration pipeline. We expect these to increase as we revise the current plan and review in more detail some of the Stage 2 and Stage 3 steps of the Calibration pipeline. From all the tests, 192 are considered basic tests, 127 classified as validation Part 1 tests and 107 classified as validation Part 2 tests. From the ones we have reviewed, 212 can be done via a computational algorithm; 201 of which are suitable to become unit tests as they don't require human intervention or inspection. The instrument teams are currently working with the development team to incorporate all unit tests into the Calibration pipeline code. Since it is publicly available via GitHub,

and therefore accessible to all science teams, we are also inspecting some of the code in order to identify other possible unit tests.

Up to now, only a small subset of the validation tests that can be coded have been identified as needing more in depth analysis after they are run. We are in the process of re-factoring our current science Calibration Pipeline Testing Tool in order to collect all these into a single validation testing framework that serves all the instrument teams and that provides with the infrastructure for a quick and complete analysis of the tests products. In order to come up with a complete set of validation tests, the science team and the development team are also organizing walkthroughs for each of the most complex steps. We are eeking to understand better how the code was implemented and where we should add some of the already identified tests.

4.1. Status

Our goal is to complete the definition of the baseline validation tests by the spring of 2019 and develop the software for all unit tests and regression tests by the end of 2019. After that, we expect to build up the validation testing as we identify new needs when inflight data becomes available. In the same time frame, we are planning to develop a set of tools to help the science team analyze and sign off on the changes to the calibration pipeline before a public release of the Calibration Pipeline software. While doing this we are also focusing developing the necessary processes to maintain consistency of the testing across all the instrument teams. We are also building a regression test suite of data that will help us validate the full range of data and science cases that are part of the core science objectives for JWST. This regression test suite will be also useful to other teams currently developing software for post calibration analysis or in preparation for commissioning of JWST.

4.2. Impact of This Approach

Scientists and developers working together to test the calibration software opened opportunities to share code and ideas that result in a better product. It reduced the resources needed for this project, provides flexibility, and allows scientists to understand better the needs of developers. This, however, has come with its challenges. It required both groups to be open to new ideas and to acknowledge the experience each group had. It also required scientists to learn about development practices and for developers to feel comfortable allowing scientists to actively participate in the software development process. It also required for us, as project leads, to actively and constantly highlight the best of both teams, such that we could succeed in spite of the different points of view and priorities.

References

STScI 2016, Jwst user documentation 2016- baltimore, md. space telescope science institute.
URL <https://jwst-docs.stsci.edu>

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Quality Assurance in the Ingestion of Data into the CDS Vizier Catalogue and Data Services

G. Landais, P. Ocirk, M. Allen, M. Brouty, E. Perret, T. Pouvreau, and P. Vannier

Centre de Données astronomiques de Strasbourg (CDS)

Abstract. Vizier is a reference service provided by the CDS for astronomical catalogues and tables published in academic journals (Ochsenbein et al. 2000), and also for associated data. Quality assurance is a key factor that guides the operations, development and maintenance of the data ingestion procedures. The catalogue ingestion pipeline involves a number of validation steps, which must be implemented with high efficiency to process the 1200 catalogues per year from the major astronomy journals. These processes involve integrated teams of software engineers, specialized data librarians (documentalists) and astronomers, and various levels of interaction with the original authors and data providers. Procedures for the ingestion of associated data (Landais 2016) have recently been improved with semi-automatic mapping of meta-data into the IVOA ObsCore standard, with an interactive tool to help authors submit their data (images, spectra, time series etc.). We present an overview of the quality assurance procedures in place for the operation of the Vizier pipelines, and identify the future challenges of increasing volumes and complexity of data. We highlight the lessons learned from implementing the FITS metadata mapping tools for authors and data providers. We show how the quality assurance is an essential part of making the Vizier data comply with FAIR (Findable, Accessible, Interoperable and Re-useable) principles, and the necessity of quality assurance in for the operational aspects of supporting more than 300,000 Vizier queries per day through multiple interactive and programmatic interfaces.

1. The Vizier staff

Vizier involves astronomers, specialized documentalists and engineers who collaborate to provide a high-quality service in conformance with the standards of the discipline. The Vizier team maintains good relations with data producers including ESO, ESA, and CNES, journals such as A&A and AAS, and collaborate with long-time partners ADS and NED.

The specialized librarians ("documentalists") collect, process and verify the data. They have constant exchanges with astronomers – for questions concerning the relevance and coherence of the data – and with IT engineers.

The engineers are in charge of the information system and application maintenance and evolution from input to the output.

The astronomers guarantee the quality of the data. Their role is to validate catalogs ingested in Vizier and to share their expertise to enhance the archives.

2. VizieR quality and curation

VizieR data come from the major astronomical journals subject to peer review and from large surveys (Gaia, Wise, LAMOST, ...) provided by trusted authorities.

The VizieR quality is the result of a dedicated expertise relying on humans and dedicated software. The process involves selecting useful data (tables, images or spectra) of scientific interest and to enhance the data with metadata describing it, as well as added value, into a VizieR catalog. Finally, VizieR provides access to these catalogs in conformance with the FAIR principle.

2.1. Data control

VizieR data control combines data consistency and data quality. Each new input are put under three levels of control.

- Format check (software): check the data conformance with the basic metadata. E.g.: data format (table or FITS), data-types, etc.
- Completeness/consistency check: data are carefully gathered and completed by documentalists. Incomplete upload is reported to authors and sometimes additional information asked (for example position missing in the tables).

Furthermore, the process allows to detect erroneous data like a nonconforming name (e.g.: SDSS J1137+2553 instead of SDSS J113723.27+255354.3), position error or value discrepancies.

When suspicious values are detected, documentalists contact authors for corrections.

- Scientific support: astronomers make themselves available to the documentalists to discuss/clarify specific problematic aspects (concepts, astronomical objects, observation techniques, units, unclear column descriptions) of the catalogs being processed. This interaction results in a better understanding of the data by the VizieR staff, which in turn improves the quality of the whole process. Astronomers are finally in charge of validating the catalog as it becomes available to the community.

2.2. The basic metadata

The table basic metadata consists of units, columns description and everything needed for the data understanding and preservation. Basic metadata also include authors, keywords and identifiers such as the VizieR identifier or the bibcode, which is shared with ADS. The first author ORCID and the Article DOI are now available, and VizieR is currently working to generate DOI for VizieR catalogs.

2.3. The rich metadata

The rich metadata are the information which enables data reuse and interoperability. The standards of the Virtual Observatory are adopted for tables (e.g. UCD) and for associated data (ObsCore Data model). Each catalog entry is customised by documentalists by adding links, positions, crossmatch and other metadata following the Virtual Observatory recommendations. This work including FITS metadata assignment as well as UCD assignment or filter detection requires specialized tools and a human expertise.

The final result is not just a catalog. It is a rich, organised collection of tables, plots, descriptions, links, associated data, and indexation available through a dedicated web application and through the Virtual Observatory. VizieR implements the VO standards such as the simple conesearch, the TAP service to access tables using ADQL/SQL or SIA, SSA and ObsTAP for associated data.

3. The curation challenge

VizieR is a crucial link between data producers and users. However, both have their specific, different demands. Data producers generate a huge data diversity, in increasing volume. The consumers are the astronomers, research pipelines or software which require data and meta-data quality that we have to ensure.

3.1. Metadata history

Since the beginning in 1996, the basic and rich metadata are assigned in VizieR.

- In 2002, the Virtual Observatory started and created standards. For instance, UCD (United Content Descriptor) constitutes a standardized nomenclature to describe columns. They are assigned for each VizieR columns.
- In 2011, VizieR started to assign photometry for magnitude in the tables.
- In 2016, associated data (spectra, images) became available through the VO interfaces - the ObsCore datamodel of the virtual observatory was chosen.
- VizieR has started to add time domain metadata and the article Identifier (DOI, ORCID).

In total, since the IVOA creation, the amount of metadata per catalog has seen a strong increase: there are over 20 potential additional metadata items and about 10 new tables of metadata among 40 tables which make up the VizieR metadata.

3.2. An important load in input

The number of articles published in journals increases linearly (see Figure 1). But more importantly, the contents of individual catalogs increase: the number of tables per catalog was multiplied by 3 between 2000 and 2018 and the number of columns per table increased from 12.8 in average in 2000 to 17 in 2018. This naturally translates to an increased workload for the documentalists who must describe and process this increasing amount of data (unit, type, description, UCD, photometry, ...). Meanwhile, the need for metadata required for interoperability also increases. For instance, the added workload of generating the metadata for the associated data and indexing the latter is particularly significant.

3.3. Lessons learned from associated data ingestion

Associated data in FITS are organized by the database generator Saada (Michel 2012). Ideally, the VizieR ingestion strategy would consist in asking the authors to generate their data's metadata themselves and sending it to the CDS via a mapping. Then the CDS validates the mapping and applies modifications if necessary.

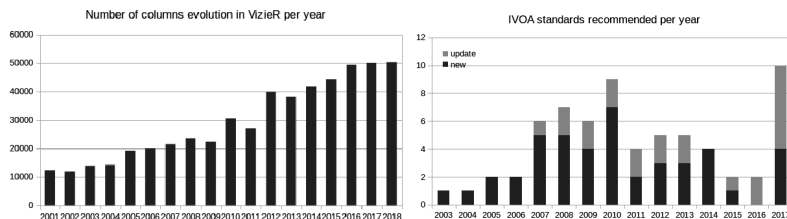


Figure 1. VizieR face to an increasing volumetry

However, this is rarely so simple, and the associated data workflow doesn't operate at full capacity. Reasons are various: this is a new data-type which needs to be assimilated by specialized documentalists. But, mostly, this is a format having trusted recommendations but which are not often followed by authors.

Currently, there have been 160 catalogue submissions using the new web interface (available in 2017 and official since January 2018) with a total of 1800 FITS files uploaded. 90% of the mapping generated by authors come from the automated process only without any user assignment. Finally, 65% of these mappings have the basic meta-data (position and spectral coordinates). Hence, the workflow must be completed by CDS, which requires manpower.

3.4. Conclusion and good initiative

- Some products are more work-intensive than others. The Associated data workflow depends on the quality of FITS generated by authors.
- Collaboration with editors and publishers facilitates the curation. For example, the XML format provided by publishers improves the workflow.
- Authors need to be educated (communication effort is needed). The recent work engaged by NED to provide a "Best Practices document" is great. The weight of the editors to ask for clean data is fundamental.
- Reference databases like filters provided by the SVO (Spanish Virtual Observatory) are useful. The recent interest in building a reference database of telescopes and instruments goes in the same direction and is awaited (Perret et al. 2018)!

This is a continuous challenge which becomes possible thanks to close collaboration with data producers and partners: journals A&A, AAS and space agencies like NASA (ADS), ESO, ESA .. and of course the astronomers' participation who submit data in VizieR.

References

- Landais, G. 2016, Mapping images and spectra metadata with ObsCore DM, Tech. rep., San Francisco
- Michel, L. 2012, A New Web Interface for Saada, Tech. Rep. 2012ASPC..461..415
- Ochsenbein, F., et al. 2000, A&AS, 143, 230
- Perret, E., et al. 2018, Shared nomenclature and identifiers for telescopes and instruments, Tech. Rep. 2018EPJWC.18604002P

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

ProvTAP: A TAP Service for Providing IVOA Provenance Metadata

François Bonnarel,¹ Mireille Louys,² Gregory Mantelet,¹ Markus Nullmeier,³
 Mathieu Servillat,⁴ Kristin Riebe,⁵ and Michèle Sanguillon⁶

¹*Université de Strasbourg, CNRS, ObAS, UMR7550, Strasbourg, France*
francois.bonnarel@astro.unistra.fr

²*Université de Strasbourg, CNRS, ICube, UMR 7357, Strasbourg, France*

³*Zentrum für Astronomie der Universität Heidelberg, Astronomisches
 Rechen-Institut, Heidelberg, Germany*

⁴*LUT, Observatoire de Paris, PSL Research University, CNRS, Meudon,
 France*

⁵*Leibniz-Institut für Astrophysik Potsdam (AIP), Potsdam, Germany*

⁶*Laboratoire Univers et Particules de Montpellier, Université de Montpellier,
 CNRS/IN2P3, Montpellier, France*

Abstract. In the astronomical Virtual Observatory, provenance metadata provide information on the processing history of the data. This is important to assert quality and truthfulness of the data, and potentially be able to replay some of the processing steps. The ProvTAP specification is an IVOA Working draft defining how to serve provenance metadata via TAP, the Table Access Protocol, which allows to query table and catalog services via the Astronomical Data Query Language (ADQL). ProvTAP services should allow finding out all activities, entities, or agents that fulfill certain conditions. Several implementations and developments are presented. The CDS ProvTAP service describes provenance metadata for the generation of HiPS image datasets. The CTA ProvTAP service will provide access to metadata describing the processing of CTA event lists. ARI-GAVO prototyped specialized query functions that could facilitate accomplishing the goals of ProvTAP users.

1. Introduction and acknowledgments

Figure 1 represents three views at different resolutions of the LMC region in the DSS2 survey using the HiPS hierarchical representation of the data. For such a data collection, it would be useful retrieving information about HiPS computation, input and output dataproducts and parameters values allowing reproducibility and quality checks. Links between progenitors (e.g. standard digital images) across the data flow and steps of processing (e.g. Schmidt plate digitization) can be described in a standard and interoperable way as proposed in the IVOA provenance data model.

Authors acknowledge support from ASTERICS project, funded by European Commission. Partial funding was also provided by GAVO, ASOV-INSU, AF CTA and PADC of the Paris Observatory. They also acknowledge Anastasia Galkin and Ole

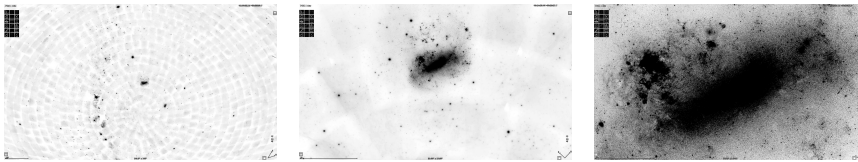


Figure 1. Views of the LMC from the DSS survey at various resolutions using the HiPS multiscale representation

Streicher from AIP for intensive discussions on the data model and careful reading of this paper.

2. The IVOA Provenance Data Model

The IVOA provenance data model has been designed to provide information on the processing history of the data. It has been derived from the W3C provenance model (Belhajjame et al. 2013) with its basic Activity, Agent and Entity classes by adding other specific classes such as Description classes allowing to group attributes that related Activities and Entities may share, or the Parameter and ParameterDescription classes which gather activity configuration features. The model is, at the time of writing, a proposed recommendation ¹ and is more detailed elsewhere in these proceedings (Servillat et al. 2019). Two IVOA DAL services specifications have been worked on for compliant provenance metadata delivery: ProvTAP (section 3) and ProvSAP ².

3. ProvTAP implementation of the Provenance Data Model

ProvTAP is an IVOA specification proposal for a TAP service (Dowler et al. 2010) delivering provenance metadata as intended by the IVOA provenance data model. The ProvTAP proposal is currently described in an internal Draft of the IVOA Data Model Working group ³. The heart of it consists of the TAP schema consisting of the descriptions of the tables mapping the classes of the model with their column metadata including datatypes, unit, ucd and utypes.

Another attempt for serializing the same metadata organization using a TripleStore has been made and is presented elsewhere in these proceedings (Louys et al. 2019).

4. The CDS HiPS provenance service content

HiPS (Fernique et al. 2017) is a VO standard describing a hierarchical organization of the data based on Healpix tessellation of the sky. CDS created 484 HiPS datasets from

¹<http://www.ivoa.net/documents/ProvenanceDM/20181015/index.html>

²at the time of writing:
<http://volute.g-vo.org/svn/trunk/projects/dm/provenance/provsap/ProvSAP.pdf>

³at the time of writing available at :
<https://wiki.ivoa.net/internal/IVOA/ObservationProvenanceDataModel/ProvTAP.pdf>

various imaging surveys and provides access to them through VO compliant applications such as Aladin or AladinLite. CDS recently created an HiPS provenance PostGres database containing metadata describing the generation of these HiPS as well as generation of some of their progenitors (for instance by digitization activities). A TAP server interface has recently been implemented on top of this database.

5. HiPS datasets Discovery examples in the CDS HiPS ProvTAP prototype service

Using the TAP service and ADQL queries it is possible to discover HiPS datasets meta-data or pointers, as well as HiPS generation activities from various VO clients, using a large variety of provenance criteria, such as:

- HiPS created by a given agent.
- Progenitors data collections of various HiPS datasets.
- HiPS generation activities or HiPS entities using configuration details. Among possible selections are:
 - all HiPS datasets for a given maximum HiPS order.
 - all HiPS datasets created for a given coordinate frame : ICRS, galactic, ecliptic, etc.
 - HiPS file format provided (jpeg, fits, etc..).

By combining entity discovery queries with more content-oriented ones using IVOA ObsCore-like descriptions of the HiPS datasets stored in a specific table, the discovered datasets can be eventually displayed in any HiPS compliant visualization software (an example using Aladin is shown on Figure 2). Other illustrations of queries and results can be found in the slides associated to the paper ⁴.

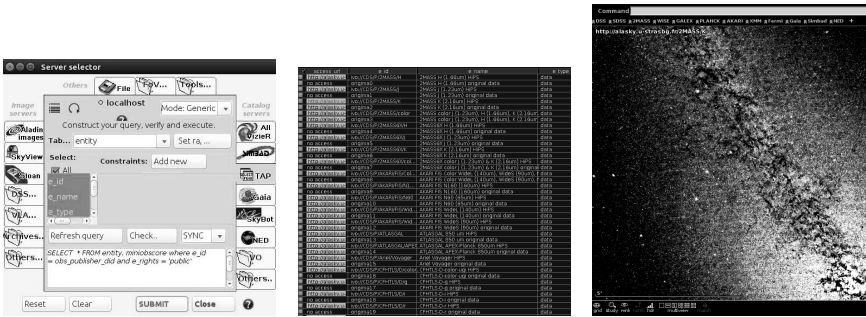


Figure 2. Left: ADQL query. Center: response join entity-ObsCore table. Right: one of the discovered HiPS datasets loaded within Aladin.

⁴<http://adass2018.astro.umd.edu/abstracts/O11-3.pdf>

6. Another ProvTAP service project : HESS/CTA

The Gamma Ray Observatory Cherenkov Telescope Array are projects where tracing the provenance is very important. The HESS telescope archive is used as a precursor of CTA for provenance. To sustain data analysis and interpretation the OPUS software⁵ used by CTA to run the processing of the data delivers provenance information by recording the processing steps, their input/output configuration and further details. This information is currently organized in a relational data base using the same TAP schema than the CDS implementation. A TAP service will be soon implemented on top of that.

7. Implementation of ADQL functions for complex queries and graph traversal

The provenance database contains fourteen tables. Selection of relevant metadata may obviously require very complex ADQL queries. In addition, provenance metadata fundamentally exhibits a graph data structure, where, e. g., progenitor entities may be found at varying distances from a resulting entity. We designed ADQL functions that cope with this complexity as a service to the ADQL user, for both usability and performance. Some of these functions, such as `find_prov_precursors()`, will return the history of a single HiPS by recursively going back through all its contributory provenance metadata. It is possible to return resulting subgraphs as a whole, using the PROV-N format (Belhajjame et al. 2013). For portability between different RDBMS backends, we made use of SQL-standardized recursive CTEs (common table expressions).

8. Conclusion

The ProvTAP specification of the IVOA DM has been implemented successfully on a PostGres database with a TAP server layer. It allows to provide provenance metadata, in an interoperable way, for HiPS generation as well as other image processing tasks (digitization of photographic plates, RGB composition, calibrations, etc..).

References

- Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Klyne, G., Lebo, T., McCusker, J., Miles, S., Myers, J., Sahoo, S., & Tilmes, C. 2013, PROV-DM: The prov data model, W3C Recommendation. URL <http://www.w3.org/TR/prov-dm/>
- Dowler, P., Rixon, G., & Tody, D. 2010, Table Access Protocol Version 1.0, IVOA Recommendation 27 March 2010. 1110.0497
- Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017, HiPS - Hierarchical Progressive Survey Version 1.0, IVOA Recommendation 19 May 2017. 1708.09704
- Louys, M., Pineau, F.-X., Bonnarel, F., & Holzmman, L. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 329
- Servillat, M., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 333

⁵<https://uws-server.readthedocs.io/en/latest/>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Rectification and Wavelength Calibration of EMIR Spectroscopic Data with Python

Nicolás Cardiel,¹ Sergio Pascual,¹ Jesús Gallego,¹ Cristina Cabello,¹
 Francisco Garzón,^{2,3} Marc Balcells,⁴ Nieves Castro-Rodríguez,⁵
 Lilian Domínguez-Palmero,^{4,2} Peter Hammersley,⁶ Nicolas Laporte,⁷
 Lee R. Patrick,² Roser Pelló,⁷ Mercedes Prieto,² and Alina Streblyanska²

¹*Departamento de Física de la Tierra y Astrofísica, Universidad Complutense de Madrid, Spain; cardiel@ucm.es*

²*Instituto de Astrofísica de Canarias, La Laguna, Tenerife, Spain*

³*Departamento de Astrofísica, Universidad de La Laguna, Tenerife, Spain*

⁴*Isaac Newton Group of Telescopes, La Palma, Spain*

⁵*Gran Telescopio Canarias, La Palma, Spain*

⁶*European Southern Observatory, Garching bei München, Germany*

⁷*Observatoire Midi-Pyrénées, Toulouse, France*

Abstract. EMIR, the near-infrared camera and multi-object spectrograph operating in the spectral region from 0.9 to 2.5 microns, has been commissioned at the Nasmyth focus of the Gran Telescopio Canarias. One of the most outstanding capabilities of EMIR is its multi-object spectroscopic mode. This work describes how important reduction steps, concerning image rectification and wavelength calibration of spectroscopic data, are performed with the help of PyEmir, the Python code developed as part of the contribution of the Universidad Complutense de Madrid in this instrument.

1. Introduction

EMIR (Garzón et al. 2016) is a near-infrared camera-spectrograph operating in the wavelength range 0.9–2.5 μ m in the Gran Telescopio Canarias that has been recently offered to the astronomical community. This instrument provides a wide range of observing modes, including imaging and spectroscopy (both long slit and multi-object, with spectral resolutions 5000, 4250 and 4000 in J, H and K, respectively).

2. The pipeline

A data reduction pipeline, PyEmir (Pascual et al. 2010, 2019b), based on Python, is being refined in order to facilitate the automatic reduction of EMIR data taken in any of the available observing modes. This package, as well as the auxiliary package Numina,

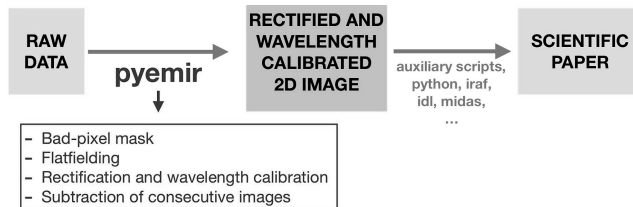


Figure 1. The Python package PyEmir helps to generate a rectified and wavelength calibrated 2D image. From this point, the astronomer can use her favorite software tools to proceed with the spectra extraction and analysis.

are both available at GitHub¹. The user's guide is being currently written after the experience gained analyzing the commissioning data, and will be soon available in the documentation-hosting platform Read the Docs. See also Pascual et al. (2019a) for a description of how to run the EMIR pipeline within a Jupyter Notebook.

3. Rectification and wavelength calibration of spectroscopic data

Focusing on the reduction of spectroscopic data (see Fig. 1), some critical manipulations are the geometric distortion correction (Figs. 2 and 3) and the wavelength calibration (Fig. 4). Using a large set of tungsten and arc calibration exposures, both calibrations have been modeled for any arbitrary configuration of the multi-object slit system. This model can be easily employed to obtain a preliminary rectified and wavelength calibrated EMIR spectroscopic image without additional calibration images (Fig. 5). This facilitates both the on-line quick reduction of the data at the telescope and the off-line detailed reduction of the data by the astronomer.

Acknowledgments. This work was funded by the Spanish Programa Nacional de Astronomía y Astrofísica under grant AYA2016-75808-R, which is partially funded by the European Development Fund (ERDF).

References

- Garzón, F., Castro-Rodríguez, N., Insausti, M., Manjavacas, E., Milucio, M., Hammersley, P., Cardiel, N., Pascual, S., González-Fernández, C., Molgó, J., Barreto, M., Fernández, P., Joven, E., López, P., Mato, A., Moreno, H., Nuñez, M., Patrón, J., Rosich, J., & Vega, N. 2016, in *Ground-based and Airborne Instrumentation for Astronomy VI*, vol. 9908 of *Proceedings of SPIE*, 99081J
- Pascual, S., Cardiel, N., & Molgó, J. 2019a, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of *ASP Conf. Ser.*, 187
- Pascual, S., Gallego, J., Cardiel, N., & Eliche-Moral, M. C. 2010, in *ADASS XIX*, edited by Y. Mizumoto, K.-I. Morita, & M. Ohishi (San Francisco: ASP), vol. 434 of *ASP Conf. Ser.*, 353
- Pascual, S., et al. 2019b, in *ADASS XXVI*, edited by M. Molinaro, K. Shortridge, & P. Pasian (San Francisco: ASP), vol. 521 of *ASP Conf. Ser.*, 232

¹<https://github.com/guaix-ucm>

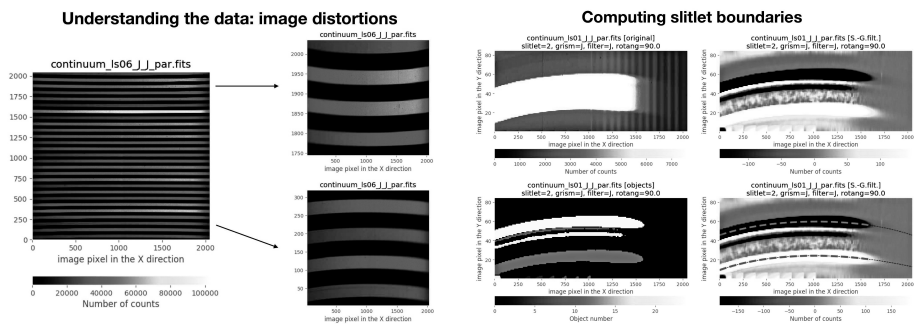


Figure 2. *Left:* The raw EMIR 2D spectroscopic images exhibit geometric distortions. This effect is clearly noticeable in this sample continuum exposure, where only even-numbered slitlets are opened (the odd-numbered slitlets where closed). *Right:* Slitlet boundaries have been computed using the continuum exposures with alternate even- and odd-numbered slitlets opened and closed. After smoothing the image to facilitate the task, the boundaries are determined using the first derivative of the signal when moving in the vertical (spatial) direction.

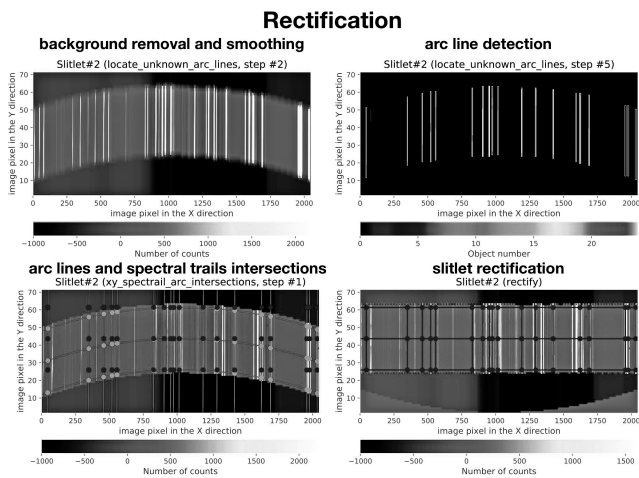


Figure 3. Once the slitlet boundaries have been determined, it is possible to rectify each 2D slitlet. In this example, an arc exposure is being analyzed in order to determine the intersections between arc lines and spectral trails. These reference points allows the computation of the rectification transformation.

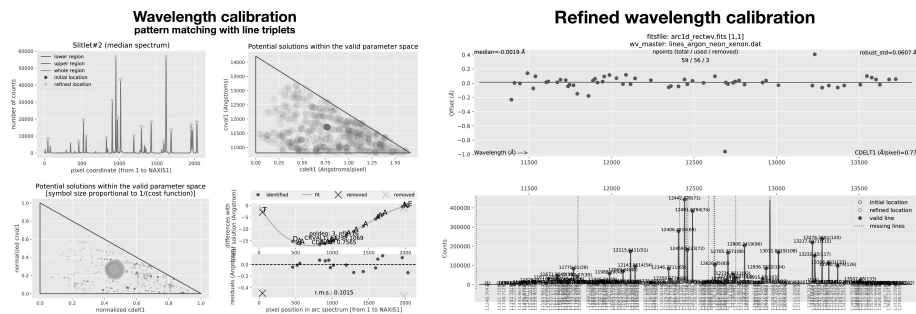


Figure 4. *Left:* After rectifying the arc-exposure slitlets, the median spectrum of each slitlet is extracted, and a pattern-matching algorithm, specially developed for PyEmir, is employed in order to derive an initial wavelength calibration solution. *Right:* The preliminary wavelength calibration is then improved by using additional (faint) arc lines, in order to achieve a typical r.m.s. of the order of 1/10 pixel.

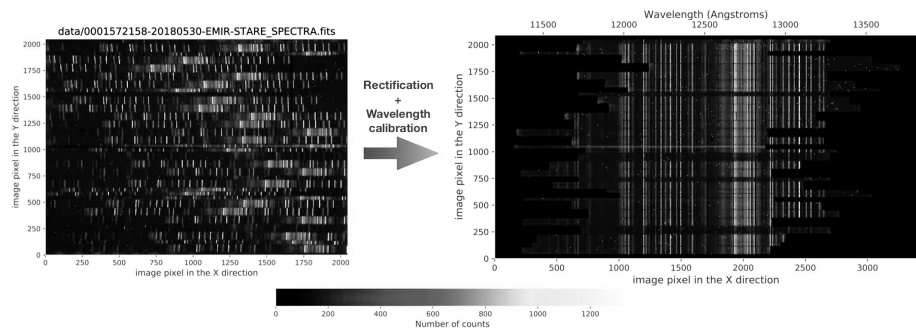


Figure 5. *Left:* Example of raw image obtained using grism J + filter J in multi-slit mode. The image exhibits the expected geometric distortions. Note that the wavelength coverage of each slitlet is different, depending on the location of the slit (bar opening) in the focal plane (cold slit unit). *Right:* The rectified and wavelength calibrated image derived using the library of calibration polynomials corresponding to the particular instrumental configuration employed.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

DRAGONS – Data Reduction for Astronomy from Gemini Observatory North and South

Kathleen Labrie,¹ Kenneth Anderson,² Ricardo Cárdenes,¹ Chris Simpson,¹
 and James E. H. Turner²

¹*Gemini Observatory, Hilo, Hawai'i, USA; klabrie@gemini.edu*

²*Gemini Observatory, La Serena, Chile*

Abstract. DRAGONS, Data Reduction for Astronomy from Gemini Observatory North and South, is Gemini's new Python-based data reduction platform. DRAGONS offers an automation system that allows for hands-off pipeline reduction of Gemini data, or of any other astronomical data once configured. The platform also allows researchers to control input parameters and in some cases will offer to interactively optimize some data reduction steps, e.g., change the order of fit and visualize the new solution. The project makes good use of other open source projects. The data interface, *Astrodata*, uses at its core Astropy's *NDData* and *io.fits*. The input parameters configuration system uses a slightly modified version of LSST's *pex.config*. The project is also working with the Astropy community to define the tools needed for building spectroscopic data reduction packages. DRAGONS is used at the observatory for nighttime quality assessment. The same software will be used for quicklook reduction of target-of-opportunity and LSST follow-up observations, and as the tool the researchers can use to prepare their Gemini data for analysis.

1. The Project

DRAGONS¹ (AURA Gemini Observatory 2018), Data Reduction for Astronomy from Gemini Observatory North and South, is a data reduction platform that includes a data abstraction layer, *Astrodata*, a process automation tool, the Recipe System, and a growing collection of data reduction algorithms. It is complemented by the Gemini Observatory Archive (GOA) software for automatic calibration association and retrieval.

DRAGONS is both an observatory system and a tool that researchers can use on their personal computer. The same codebase is used for both purposes, displaying great flexibility and minimizing long-term maintenance cost.

Currently, the main project objectives are:

- Data reduction tools and pipeline for the user **and** the observatory.
- Nighttime quality assessment of the data.
- Automated reduction for quick nighttime evaluation of LSST follow-up.

¹ Code: <https://github.com/GeminiDRSoftware/DRAGONS>; Documentation <https://dragons.readthedocs.io/>.

DRAGONS, along with the Gemini Observatory Archive, will be central to Gemini’s LSST follow-up system by reducing imaging and longslit data at night and feeding the products back into the system for evaluation by the researchers. Already, DRAGONS is run at night to produce quick reduction of any imaging data and to produce sky condition metrics to help guide the queue observer.

The infrastructure components of DRAGONS, Astrodata, and the Recipe System, are completed, as well as most routines required for the reduction of imaging data from all current facility instruments (Labrie et al. 2019). The development team is now focusing on implementing algorithms for spectroscopy. All future Gemini instruments, e.g., GHOST (Ireland et al. 2018) and Scorpio, will be using DRAGONS for data reduction.

2. Astrodata

Astrodata is DRAGONS’ data abstraction layer to represent datasets stored on disks. Currently, it implements a representation for MEF file, though Astrodata is fully extensible to other formats. The AstroData class uses the NDData class as its core representation, as opposed to, for example, the HDUlist provided by pyfits (Figure 1).

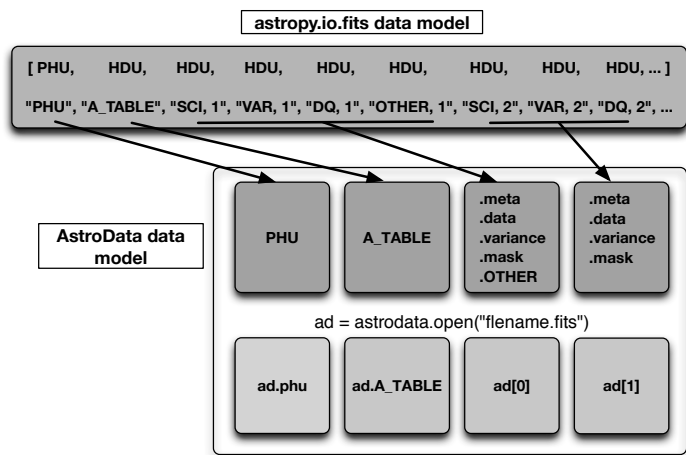


Figure 1. Astrodata currently has a layer to read in a Multi-Extension FITS file and convert it into its internal representation. That internal representation uses the NDData model.

Astrodata provides uniform interfaces for working on data from different instruments. It also provides a number attributes used by DRAGONS’ Recipe System to associate data reduction recipes and primitives. The `tags` attribute is the central key to the mapping as it lists the core characteristics of the data, eg., the instrument name, whether it is imaging or spectroscopy data, whether it is a science frame or a calibration frame, if the latter, of which type, etc. The `descriptors`, a set of methods, serve as header access and keyword mapping. All the details specific to an instrument are coded inside the class associated with the instrument. That class then provides the interface.

The appropriate class is selected automatically when the file is opened and inspected by Astrodata.

3. Recipe System

The Recipe System is launched either from a command line interface called “reduce” or programmatically through an API, specifically, the class “Reduce.” The Recipe System uses an external calibration service to request processed calibration matching the data being reduced. At the Observatory, the calibration service accesses an internal Postgres database via FitsStorage, the software used to drive and access the public GOA. In a researcher’s environment, a simple desktop or laptop, the calibration service accesses an sqlite database with a simple command-line or API interface. The researcher version is a subset of the full GOA software and uses exactly the same calibration association rules as the Gemini archive.

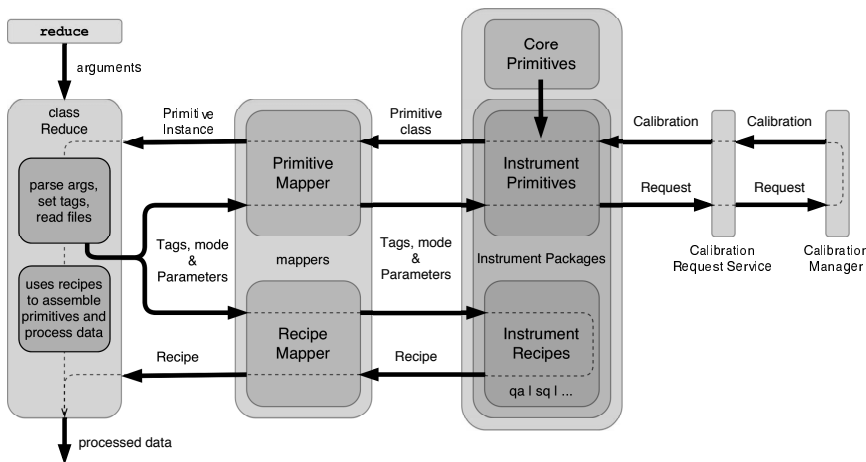


Figure 2. The schematic of all of the Recipe System components.

The Recipe System feeds an AstroData object to the mappers to find the most appropriate recipe and primitives for the input data, as well as the appropriate default input parameters for each primitive. The association is based on AstroData tags (Section 2) and operation mode (e.g., science quality, sq, nighttime sky condition monitoring, qa.). The input parameter system is a slightly modified version of LSST’s `pex.config` package. Default parameters can be set per instrument, wavelength regime, etc., and, like the primitives, can be inherited thus avoiding unnecessary code duplication. The tool also allows researchers to modify those input parameters and customize a reduction. Figure 2 shows a full schematic of the Recipe System.

4. Conclusion

DRAGONS is publicly available on Github at:

<https://github.com/GeminiDRSoftware/DRAGONS>

After the first public release, DRAGONS is expected to be operated as an Open Source project, welcoming community fixes, though Gemini will continue to provide full support for continuing development with a focus on the Observatory's needs.

Acknowledgments. The Gemini Observatory is operated by the Association of Universities for Research in Astronomy, Inc., under a cooperative agreement with the NSF on behalf of the Gemini partnership: the National Science Foundation (United States), the Science and Technology Facilities Council (United Kingdom), the National Research Council (Canada), CONICYT(Chile), Ministério da Ciência e Tecnologia (Brazil), and Ministerio de Ciencia, Tecnología e Innovación Productiva (Argentina)

References

- AURA Gemini Observatory, S. U. S. D. 2018, DRAGONS, Astrophysics Source Code Library. 1811.002
- Ireland, M. J., White, M., Bento, J. P., Farrell, T., Labrie, K., Luvaul, L., Nielsen, J. G., & Simpson, C. 2018, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 10707 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 1070735
- Labrie, K., Cárdenes, R., Anderson, K., Simpson, C., & Turner, J. 2019, in ADASS XXVII, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 999 TBD



CMNS Dean Amitabh Varshney opening ADASS telling us how Astronomy has inspired his field of Visual Computing. (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

stginga: Ginga Plugins for Data Analysis and Quality Assurance of HST and JWST Science Data

Pey Lian Lim¹ and Eric Jeschke²

¹*STScI, Baltimore, MD, USA; lim@stsci.edu*

²*NAOJ, Hilo, HI, USA*

Abstract. `stginga`¹ is an image visualization package to assist in data analysis and quality assurance of science data from Hubble Space Telescope (HST) and James Webb Space Telescope (JWST). It is based on the `Ginga`² toolkit for building scientific viewers. In this article, we will describe the main plugins developed with `stginga`. We also discuss the basic outline of writing a `Ginga` plugin, with pointers to documentation and examples.

1. Introduction

`Ginga` (Jeschke et al. 2015) is a Python package that implements a toolkit for building scientific viewers. It provides a *reference viewer*, which features a plugin architecture in which nearly every graphical feature of the program is implemented by a Python plugin. By implementing some new plugins for the HST and JWST data analysis and quality assurance tasks, and combining these with a curated selection of the distributed “stock” plugins, we were able to fairly quickly develop a tool for use in the HST and JWST community.

The reference viewer separates image data into virtual holding pens called *channels*, named and organized by the user. Plugins are categorized as *global* or *local*. A global plugin applies to all images across all channels: only one instance can be opened in the whole `Ginga` session, whereas a local plugin is associated with the channel it is started from: one instance can be opened per channel and different instances can be configured separately in the same `Ginga` session.

2. BackgroundSub

BackgroundSub (see Figure 1) is used to calculate and subtract background value. User draws a shape (e.g., annulus) to define the region from which background is calculated. As user modifies the region or changes the parameters in the “Attributes” box, background value would be recalculated accordingly. Optionally, if a data quality (DQ)

¹<https://github.com/spacetelescope/stginga> (STScI)

²<https://github.com/ejeschke/ginga> (NAOJ)

extension is available, pixels marked as “not good” also can be excluded from calculations. Subtraction parameters can be saved to a JSON file, which then can be reloaded.

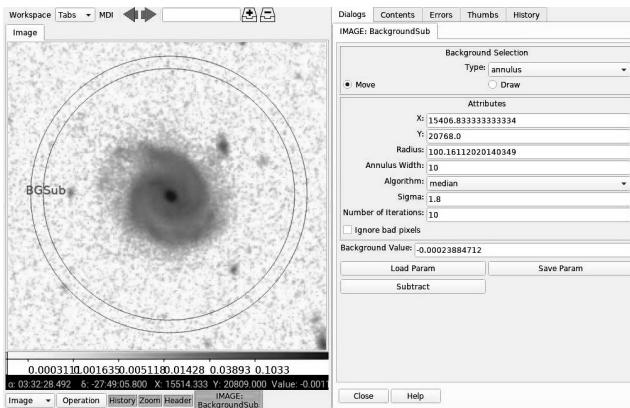


Figure 1. BackgroundSub plugin for background subtraction.

Then, the calculated background can be subtracted off the displayed image in Ginga. However, the subtracted image only exists in an in-memory cache in Ginga; if the cache fills up Ginga will eject the image if it is not being viewed. To save the subtracted image out to a different file, use the *SaveImage* plugin in Ginga. As of this writing, *BackgroundSub* only handles a constant background, therefore it is unsuitable when the background has a gradient or a pattern.

3. BadPixCorr

*BadPixCorr*³ is a plugin for performing interactive bad pixel correction on an image.⁴ Currently, it only handles fixing a single bad pixel or bad pixels within a circular region. The bad pixel(s) can be filled either by a user-defined constant, a constant calculated from an annulus (not unlike *BackgroundSub*), or Scipy griddata interpolation using the annulus. If DQ extension is present, the corresponding DQ flags will also be set to the given new flag value (default is 0 for “good”).

4. DQInspect

DQInspect (see Figure 2) is used to visualize the associated DQ array stored as an HDU within an image. It shows the different DQ flags (top table) that went into a selected pixel (marked by a red “x”) and also the overall mask of the selected DQ flag(s) (blue/covered pixels; bottom table). For overall mask, when multiple flags are selected, each flag is assigned a different mask color at a reduced opacity for each. User has the option to customize flag definitions for different instruments.

³Figure not shown here but available in the corresponding poster.

⁴See *BackgroundSub* for comments on JSON support and in-memory cache handling of corrected image.

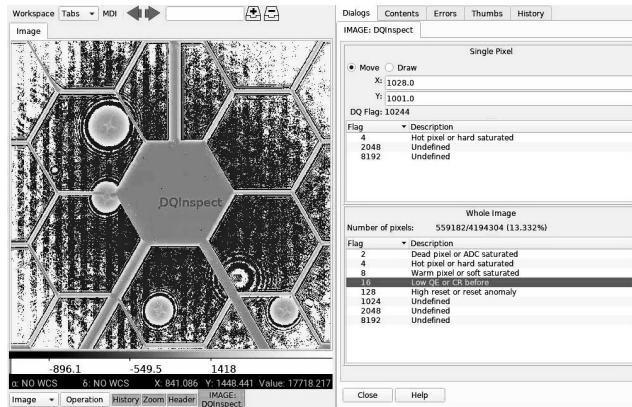


Figure 2. DQInspect plugin for data quality inspection.

5. SNRCalc

*SNRCalc*³ is used to calculate Signal-to-Noise Ratio (SNR) and Surface Background Ratio (SBR) on an image. Given the selected science (S) and background (B) regions, SBR is defined by Ball Aerospace (Acton 2015) as the median of S divided by the standard deviation of B . If the image has an accompanying error (E) extension, SNR can also be calculated by dividing S by E over the same region and then computing its minimum, maximum, and mean.

While SNR is more popular, SBR is useful for an image without existing or reliable error values. Users may define a minimum limit for SBR check, so that the GUI can provide a quick visual indication on whether the selected region achieves the desired SBR or not. As part of the statistics, mean background value is also provided albeit not used in SBR nor SNR calculations. Optionally, if DQ extension is available, pixels marked as “not good” can be excluded from calculations as well. Calculated values can be saved in the image header using the “Update HDR” button.⁴

6. Writing a Ginga plugin

Instructions for writing a plugin is available at <https://bit.ly/writeplugins>. Existing plugins in Ginga and stginga code repositories can be used as examples. It is recommended that you play with the existing ones and choose one that is the closest to your desired functionality as a starting point.

6.1. Local plugins

A local plugin at its simplest is just a Python class defined in a file. The class should inherit from *ginga.GingaPlugin.LocalPlugin* and provide *__init__()*, *build_gui()*, *start()*, and *stop()* methods. These methods are used to initialize the plugin, build the user interface, and to do any necessary tasks at the start and stop of the plugin, respectively. Typically you would also want to implement the *redo()* method, which is called when there is new data loaded into the viewer to which the running plugin should respond.

Inside the file, any modules that are available in the user's Python environment may be imported and used, allowing huge flexibility in the kinds of things a plugin can do. It can open files, connect to sockets or other communication frameworks, or call a myriad of astronomical Python packages. It also has a reference to the viewer with which it is associated so it can access the viewer data (as a Numpy array) and can manipulate canvas overlays with graphics on the viewer (as shown in the sections above) or manipulate the viewer settings (e.g., panning, scale, color map).

6.2. Global plugins

Writing a global plugin is similar to the process for writing a local one. The difference is that the plugin ostensibly must be able to update its state when the user switches channels, since there only one instance of the plugin is allowed to be open; There are callbacks for which you can register to be alerted of these events. Otherwise, the API is quite similar to that of a local plugin.

6.3. Distributing plugins

When you want to distribute your plugin(s) the best way is to probably use the *ginga-plugin-template*.⁵ This template allows one or more plugins to be installed as a separate package, and be discovered by the reference viewer when it starts up. If you want more control over the layout of the viewer and the set of included plugins, you can follow the path blazed by *stginga* and make your own startup script for the reference viewer with a curated mix of the stock plugins with your own.

7. Conclusion

stginga utilizes Ginga plugins to support HST and JWST data analysis, which includes background subtraction, bad pixel correction, DQ flags inspection, and signal-to-noise calculations.

Writing Ginga plugins can be an expedient way to develop graphical data analysis and quality assurance tasks, by leveraging the combination of Python, a lean Ginga plugin API, and the burgeoning number of open-source astronomical Python modules.

Both *stginga* and *ginga* are installable via *pip*. Alternately, if you use *conda*, they are also available on *AstroConda*,⁶ in addition to *ginga* being in *conda-forge* too.

References

- Acton, S. 2015, Image Pre-Processor SBR, Private communications
Jeschke, E., Inagaki, T., & Kackley, R. 2015, in *Astronomical Data Analysis Software and Systems XXIV*, edited by A. R. Taylor, & E. Rosolowsky (ASP). Vol. 495

⁵<https://github.com/ejeschke/ginga-plugin-template> (NAOJ)

⁶<https://astroconda.readthedocs.io> (STScI)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

A Triplestore Implementation of the IVOA Provenance Data Model

Mireille Louys,^{1,2} François-Xavier Pineau,² François Bonnarel,² and
Lucas Holzmann³

¹*ICube UMR 7357-CNRS, University of Strasbourg, France;*
mireille.louys@unistra.fr

²*CDS, Observatoire astronomique UMR 7550-CNRS, University of*
Strasbourg, France

³*ENSIIE, EVRY, France*

Abstract. The IVOA (International Virtual Observatory Alliance) has proposed a standard for capturing the provenance metadata in the production and distribution of astronomical data. We present an implementation in a triplestore for the provenance information recorded for a collection of astronomical images. The ontology applied is derived from PROV-O from the W3C and from the IVOA Provenance data model. SPARQL queries based on the data model concepts allow to select datasets on a wide range of provenance properties and have proven to be efficient in the triplestore representation. The data model of the SIMBAD CDS database has also been tested, and turned out to scale very efficiently in the triplestore strategy as well.

1. Goal

This paper presents an evaluation of a triplestore implementation and its comparison with a relational database implementation. The two databases serve provenance metadata as modeled following the IVOA Provenance data model available at <http://www.ivoa.net/documents/ProvenanceDM/>. We present how the data model has been translated to an equivalent ontology and how table data from a Prov-TAP service can be translated to a list of triples following the semantics of the ontology. Equivalent queries have been addressed to both database architectures in order to test expressivity and extensibility as well as completeness for both systems. This exercise is also an experiment on the way the data model classes and relations are put in practice and activated with real datasets. It helped to understand how relations, roles and properties can be enhanced in the data model implementation.

2. Provenance Metadata representation for an astronomical image database

The IVOA Provenance data model is an IVOA specification for structuring and describing metadata about the history of data sets preparation and publication in astronomy. It

is elaborated as a deliverable for DADI provenance effort in the ASTERICS project¹ in connection with main astronomical projects like CTA². It represents the operations applied to data, the agent performing or responsible for these operations and the entities, typically data sets and parameters involved in these applications.

3. Implementations of the IVOA Provenance DM

The typical implementation of the IVOA provenance DM is through a relational database architecture as done for the RAVE provenance use case, CTA pipe project, CDS test image database and planned for the SVOM project. In this design, each class of the data model is stored as one database table, with each row representing an instance (e.g. an Entity instance representing a data set is stored in the Entity table, an agent instance in the Agent table, etc.). Relations between instances are stored as instances (rows) in the corresponding relation table (e.g. 'Entity E1 wasGeneratedBy Activity A1' is stored as a row in the wasGeneratedBy table and binds both identifiers from E1 and A1.)

Another way to represent those metadata is to use RDF/ttl in a triplestore representation, gathering all properties we know about the instances in sentences built on an ontology that translates the relations and attributes defined in the model. As the IVOA Provenance DM extends the W3C PROV-DM, we extended the W3C PROV-O ontology ((Belhajjame et al. 2013)) to tackle the extended part of the model : description, configuration with parameter, etc. The draft ontology is available at <http://wiki.ivoa.net/internal/IVOA/ProvenanceRFC/provOntologyIVOA2018-v1-0.owl> and sketched in Fig. 1.

3.1. The CDS test image database

We have gathered a subset of images processed for feeding the Aladin image database server in order to setup a prototype database for tracing provenance metadata from image digitization, image cutouts, RGB image combination, as well as computation of HiPS datasets. The CDS test database is available via a TAP service, following the IVOA TAP protocol ((Dowler et al. 2010)) built-up on a PostgreSQL database backend.

3.2. Triplestore implementation for CDS test data

The BlazeGraph (<https://www.blazegraph.com>) platform was chosen as a testbed in order to evaluate the merits of a triplestore for the representation of our CDS test database. All classes and relations have been translated from database tables to VOTable then in CSV lists, one for each class or relation. The ingestion in the triplestore builds one triple for each instance of a class or of a relation. Every relation in the model is translated in a distinct unique predicate. Roles for Agents need to be created with an extra predicate (holdsRoleInTime) in order to allow one Agent instance to play different roles with respect to an Activity or an Entity.

¹<http://www.asterics2020.eu>

²<https://www.cta-observatory.org>

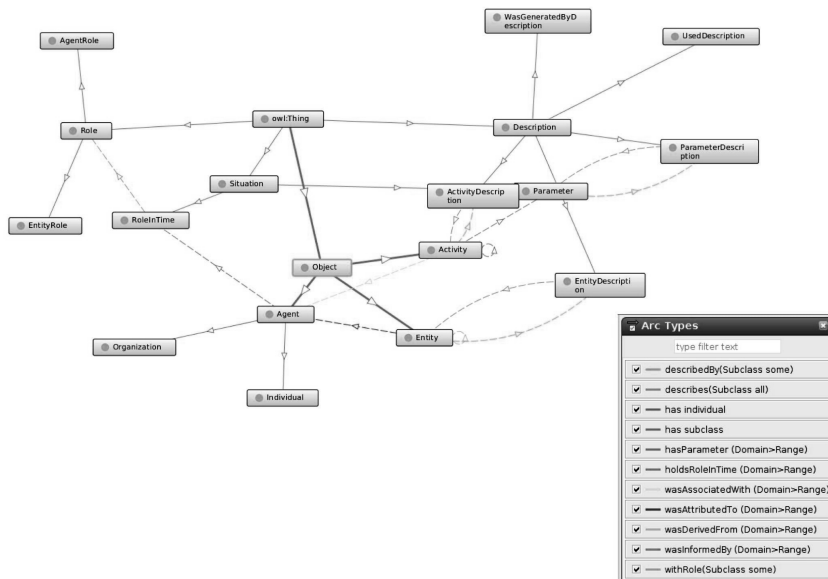


Figure 1. Provenance Ontology extending the W3C PROV-O and supporting the IVOA Provenance DM.

At the end of the translation, the triplestore entails the same provenance metadata as the CDS prototype PostGres database and implements the following set of classes and relations :

Table 1. Provenance Metadata supported in the CDS triplestore implementation

Classes	Relations
Activity	hadDescription
Activity	hadConfiguration
Entity	used
	wasGeneratedBy
	wasDerivedFrom
Agent	wasAttributedTo
	wasAssociatedTo
ActivityDescription	hadDescription
Parameter	hadDescription
ParameterDescription	
UsedDescription	
WasGeneratedByDescription	

The Blazegraph CDS repository is described as a reference implementation at the IVOA Provenance DM review page <https://wiki.ivoa.net/twiki/bin/view/IVOA/ProvenanceRFC>.

3.3. Triplestore queries for evaluation on CDS test data

A list of test queries has been defined in order to evaluate the capabilities of the triplestore queries to match with those of the Postgres/TAP service. This is listed on the Provenance DM RFC page at

<http://wiki.ivoa.net/internal/IVOA/ProvenanceRFC/ProvQuerytest-3store.pdf>.

All queries addressed to the TAP service have a translated version in the triplestore implementation and provide a similar list of hits. The equivalence of representation is confirmed and the triplestore offers at least the same expressivity for queries. The query mechanism uses name matching for relations and for class attributes. It filters the triples on the discovered values. No complex nor intricate joins are required. More investigation is needed to check how the filtering can be ordered to optimize the search.

4. Lessons Learned

The triplestore is more efficient if relations are specialized on the various classes. Partitions in various types of entities, namely Parameters and EntityData help to speed-up the search. Relations in the model can be qualified by attributes (the typical association class in UML). In the triplestore this requires an additional predicate. As an example, the relation "wasAttributedto" between an Entity and Agent has an extra predicate : 'holdsRoleInTime' which allows for an Agent to play various roles with respect to this entity: operator, author, provider, etc.

The number of triples is not limited and can be increased easily with the number of instances. The triple is a flat and additive representation and can be used to extend the model, for instance by adding new properties via new attributes to classes or to relations, or adding new relations between existing class instances, etc. Then tracing the evolution of provenance metadata in a project is available and at the same time, the compatibility to current IVOA Provenance data can co-exist.

A scaling test has been performed on Blazegraph implementing an excerpt of the CDS SIMBAD database with a set of 20 relations. Successive increase of the number of astronomical SIMBAD objects showed that Blazegraph has scaled properly for 10 000, 1million and 8,5 millions of objects. In order to speed-up the positional search for those objects a positional index based on HEALPIX coordinates representation has been successfully applied.

Other triplestore platforms will be tested to complete the tests.

Acknowledgments. This work has been sponsored by the DADI work package from the ASTERICS project (EU Horizon 2020) and the CDS internship program.

References

- Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., & Zhao, J. 2013, Prov-o: The prov ontology, W3C Recommendation. URL <http://www.w3.org/TR/prov-o/>
- Dowler, P., Rixon, G., & Tody, D. 2010, Table Access Protocol Version 1.0, IVOA Recommendation 27 March 2010. 1110.0497

The IVOA Provenance Data Model

Mathieu Servillat,¹ Kristin Riebe,² François Bonnarel,³ Anastasia Galkin,²
Mireille Louys,^{3,4} Markus Nullmeier,⁵ Michèle Sanguillon,⁶ and Ole Streicher²

¹*Laboratoire Univers et Théories, Observatoire de Paris, PSL Research University, CNRS, 92190 Meudon, France; mathieu.servillat@obspm.fr*

²*Leibniz Institute for Astrophysics Potsdam, Germany*

²*Centre de Données astronomiques de Strasbourg, Observatoire Astronomique de Strasbourg, Université de Strasbourg, CNRS, Strasbourg, France*

³*ICube Laboratory, Université de Strasbourg, CNRS, Strasbourg, France*

⁴*Zentrum für Astronomie der Universität Heidelberg, Astronomisches Rechen-Institut, Heidelberg, Germany*

¹*Laboratoire Univers et Particules de Montpellier, Université de Montpellier, CNRS/IN2P3, France*

Abstract. A provenance data model is currently discussed within the International Virtual Observatory Alliance (IVOA). The objective is to describe how provenance information can be modelled, stored and exchanged within the astronomical community in a standardized way. We follow the definition of provenance as proposed by the World Wide Web Consortium (W3C), i.e. that "provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness". Such provenance information in astronomy is important to enable any scientist to trace back the origin of a dataset (e.g. an image, spectrum, catalog or single points in a spectral energy distribution or a light curve), a document (e.g. an article, a technical note) or a device (e.g. a camera, a telescope), learn about the people and organizations involved in a project and assess the usefulness of a dataset, document or device for her own scientific work. In the astronomy domain, the user generally requires additional information on the activities, in the form of description, configuration and context information.

1. Objectives and requirements

The IVOA Provenance data model (Servillat et al. 2018; Riebe et al. 2019) shall capture information in a machine-readable way that would enable a scientist who has no prior knowledge about a dataset to get more background information. This will help the scientist to decide if the dataset is adequate for her research goal, assess its quality and get enough information to be able to trace back its history as far as required or possible.

The Provenance data model should help to solve the following tasks:

- Find out which steps were taken to produce a dataset and list the methods, tools or software that were involved,

- Find the people involved in the production of a dataset, the people, organizations or institutes that need to be cited or can be asked for more information,
- Find the location of possible error sources in the generation of a dataset,
- Judge the quality of an observation, production step or dataset,
- Search for specific products in structured provenance metadata.

2. Core and extended model

The core of the model is derived from the W3C PROV data model (Belhajjame et al. 2013) that provides a generic structure to trace the lineage of any *entity*, through its relations (e.g. *generation* and *usage*) with *activities* and *agents*. A general example of a chain tracing back the origin of released data is shown in Figure 1.

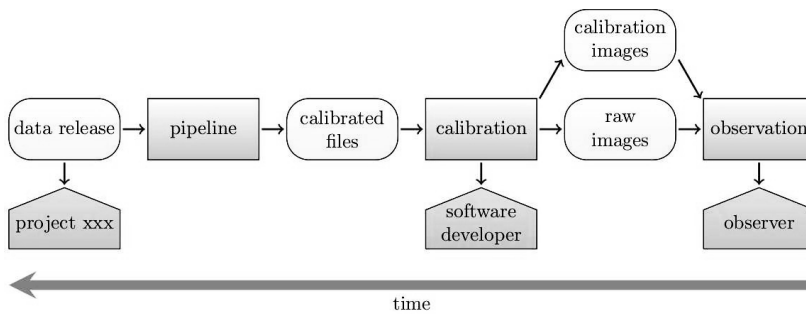


Figure 1. An example graph of provenance discovery.

In astronomy, entities are generally datasets composed of VOTables, FITS files, database tables or files containing values (spectra, light curves), logs, parameters. The activities correspond to processes like an observation, a simulation, processing steps (image stacking, object extraction, etc.), execution of data analysis code, publication, etc. The people involved can be for example individual persons (observer, publisher, etc.), groups or organisations.

In the extended model (see Figure 2), the core model is fleshed with provenance information that is relevant in the astronomy domain:

- *How was the calibration performed, which steps, which algorithms?*
Description: information on the expected working of an activity and on the expected structure of an entity. This descriptive information is what is known before any activity or entity instance is created,
- *What was the detailed configuration of the pipeline?*
Configuration: information passed to an activity in order to configure its execution and which directly influences the development of the activity (e.g. Parameter, Config File, ObsConfig),
- *What was the weather during the observation, which hardware was used?*
Context: information on the context that influences the development of an activity, but for which there is no or little control at the moment of its execution (e.g. Ambient Conditions, Instrumental Context, Execution Environment).

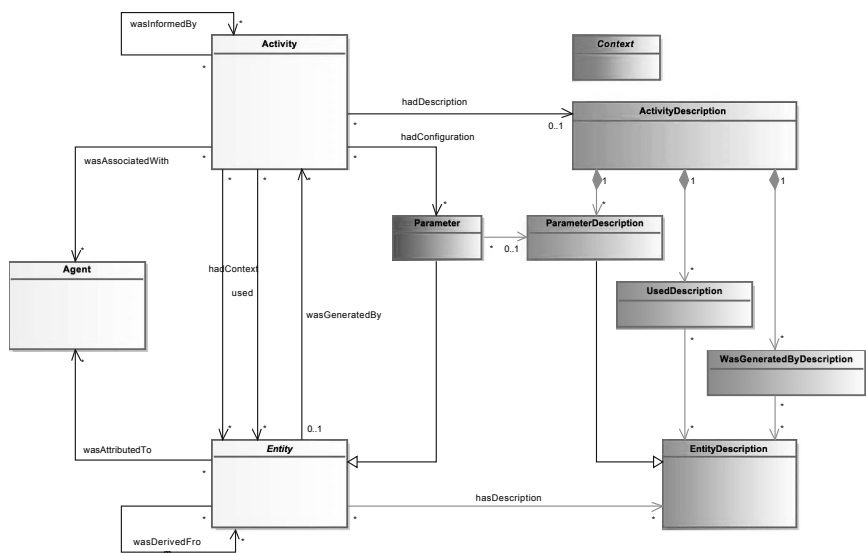


Figure 2. Class diagram showing the main functional features of the ProvenanceDM. This diagram includes only a subset of all the specialized entities and specialized relations which are further presented in diagrams 3 and 4.

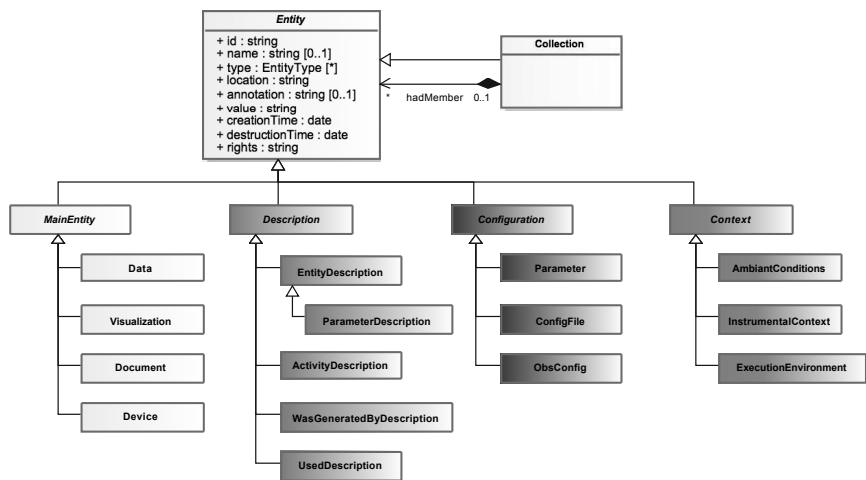


Figure 3. Class diagram showing the structuring links for specialized entities.

Specialized entities are defined and presented in Figure 3, following the categories listed above. Specialized relation are defined correspondingly in Figure 4.

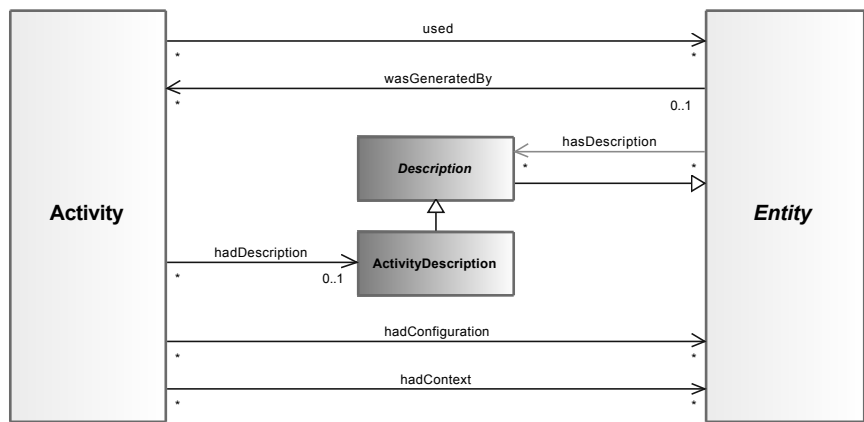


Figure 4. Class diagram showing the relations between specialized entities and Activity or Entity.

Acknowledgments. We acknowledge support from the Astronomy ESFRI and Research Infrastructure Cluster - ASTERICS project¹, funded by the European Commission under the Horizon 2020 Programme (GA 653477). This document has been developed in part with support from the German Astrophysical Virtual Observatory, funded by BMBF Bewilligungsnummer 05A14BAD and 05A08VHA. Additional funding was provided by the INSU (Action Spécifique Observatoire Virtuel, ASOV), the Action Fédératrice CTA at the Observatoire de Paris and the Paris Astronomical Data Centre (PADC).

References

Belhajjame, K., B’Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Klyne, G., Lebo, T., McCusker, J., Miles, S., Myers, J., Sahoo, S., & Tilmes, C. 2013, PROV-DM: The prov data model, W3C Recommendation. URL <http://www.w3.org/TR/prov-dm/>

Riebe, K., Bonnarel, F., Louys, M., Rothmaier, F., Sanguillon, M., & Servillat, M. 2019, in ADASS XXVI, edited by M. Molinaro, K. Shortridge, & P. Pasian (San Francisco: ASP), vol. 521 of ASP Conf. Ser., 453

Servillat, M., Riebe, K., Bonnarel, F., Galkin, A., Louys, M., Nullmeier, M., Sanguillon, M., & Streicher, O. 2018, IVOA provenance data model, <http://www.ivoa.net/documents/ProvenanceDM/>

¹<http://www.asterics2020.eu/>

Session VII

DevOps Practices in Astronomy Software

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

DevOps: A Perfect Ally for Science Operations for Large and Distributed Astronomy Projects like Gaia

Rocio Guerra,¹ Neil Cheek,² Eduardo Anglada,³ Pilar Esquej,⁴ Emilio Fraile,⁴ Enrique Pozo⁵, and Uwe Lammers¹

¹*European Space Agency (ESA), ESAC, Madrid, Spain;*
rguerra@sciops.esa.int

²*Serco Gestion de Negocios S.L. for ESA, ESAC, Madrid, Spain*

³*ATG Europe for ESA, ESAC, Madrid, Spain*

⁴*RHEA for ESA, ESAC, Madrid, Spain*

⁵*Aurora Technology B.V. for ESA, ESAC, Madrid, Spain*

Abstract. The Gaia Science Operations Centre (SOC) is an integral part of a large consortium responsible for the Gaia data processing task. Serving terabytes of processed data on a daily basis to other Processing Centres across Europe makes unique demands on the processes, procedures, as well as the team itself. This paper describes how we have embraced the DevOps principles to achieve our goals on performance, reliability and teamwork.

1. Introduction

Gaia was launched on December 19th 2013 with an ambitious objective: to create the largest, most precise three-dimensional map of the Milky Way in order to reveal its content, dynamics, current state and formation history. The second data release (Gaia DR2) delivered on April 25th 2018 is the most recent proof of the success of the mission and the revolution this largest catalogue is (already) bringing to practically all fields of astronomy.

The Data Processing and Analysis Consortium (DPAC) was entitled in 2006 to carry out the processing of Gaia's data with the final objective of producing the Gaia Catalogue. That is a very complex undertaking that encompasses many different tasks: developing the data processing algorithms and corresponding software, providing the IT infrastructure and executing those during the mission. The end-to-end processing starts with the raw data received from Gaia and culminates with the generation of the final scientific data products that will be released to the scientific community.

DPAC is a large entity of more than 160 institutes and ESA involving around 450 scientists and experts in the fields of software development and engineering. DPAC is organized in specialized groups (for astrometry, photometry, variability, etc.) known as Coordination Units (CUs) in charge of the design of the scientific algorithms and their software implementation and the Data Processing Centres (DPCs) where those processing software systems are integrated and executed within a suitable IT infrastructure.

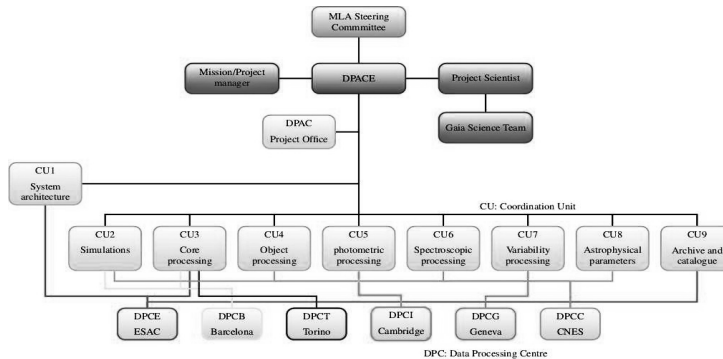


Figure 1. DPAC organigram

Fig. 1 shows the DPAC organigram depicting the Coordination Units and Data Processing Centres (plus the DPAC management bodies at the top). Coloured lines connect each CU with the DPC that runs its software (O’Mullane 2014). The Gaia Science Operations Centre is an integral part of the DPAC consortium. It performs additional roles to those generally assigned to a “classical” ESAC SOC, mainly:

- be the Mission Operations Centre (MOC) primary contact for all the payload and science-related mission aspects,
- produce the mission planning products for MOC (Scan Law, Science Schedule, etc.),
- perform regular payload health monitoring,
- process first-level products and disseminate them downstream and
- develop of the mission archive and its operations.

In addition, the SOC acts as a Data Processing Centre (the DPCE) for DPAC and is home to Coordination Unit teams: CU1 (defining system architecture and developing and running the DPAC central repository – Main Database), CU3 (producing daily intermediate products and developing and operating the astrometric core system – AGIS) and CU9 (supporting the catalogue creation). Fig. 2 highlights (in red circles) the DPAC activities performed at ESA. Excluding the operations of the archive, the SOC data processing is divided in cyclic and daily activities.

The cyclic processing (AGIS, the Astrometric Global Iterative Solution) achieves the ultimate accuracy through iterations. Therefore, it is run several times during the mission incorporating more and more observation data and new improved results from other DPAC systems.

On the other hand, there is a processing pipeline continuously up and running as per a data driven approach, i.e. any time MOC pushes new acquired science telemetry from Gaia ($\approx 35\text{--}100\text{ GB/day}$), it enters into the pipeline that unpacks and decompresses the raw inputs, determines basic image parameters, cross matches the observations taken during the day with entries in the reference catalogue and monitors the health of the payload. The main results ($\approx 300\text{--}500\text{GB/day}$) are stored in the central

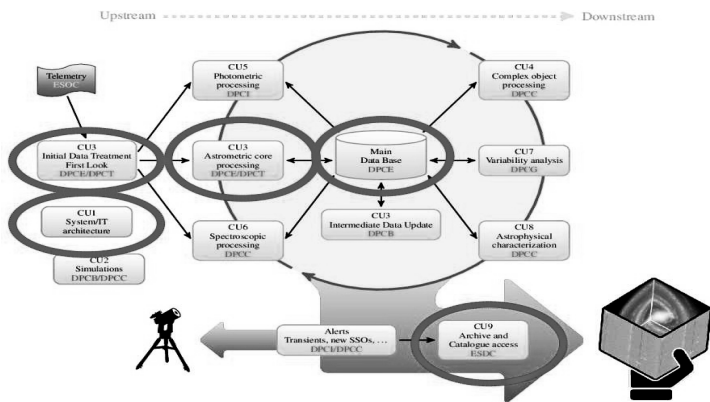


Figure 2. ESA’s contribution to DPAC

repository – the Main Database – and distributed to the other Data Processing Centres (Fig. 3). This paper describes the challenges that this pipeline imposes to the Operations team and the solutions adopted in order to achieve a smooth sustained processing.

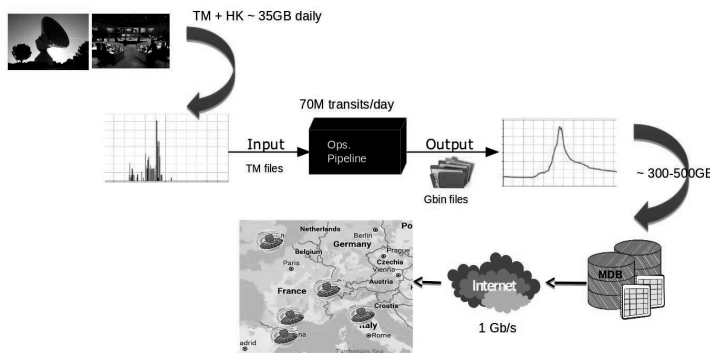


Figure 3. Schematic overview of the SOC Daily Pipeline

2. Challenges

The daily pipeline running at the Gaia SOC is very complex as it was designed to fulfill stringent requirements especially in terms of performance (Siddiqui 2014). The reason is twofold. Firstly, DPAC produces science alerts (photometry and solar system alerts) in other Data Processing Centres that need data as quickly as possible. Secondly, the payload health monitoring aims to detect and diagnose possible onboard issues as soon as possible.

Both needs require that the processing is done in near real-time. Because the telemetry does not arrive time-ordered (it follows a priority downlink scheme - plus

there may be issues that corrupt the incoming data) the integrated systems were designed to be very tightly coupled with many dependencies among them in order to produce small time ordered chunks of data to be distributed downstream.

On the other hand, the volume of data to handle is massive (Hernandez & Hutton 2015). The average size of telemetry (science and housekeeping) received from MOC daily is 35GB (70TB in total since launch). And for denser areas of the sky scanned by Gaia (like the Galactic plane) it increases up to 90–100 GB. The outputs stored in the Main Database are about 300–500GB/day (and most of them are disseminated to the rest of the DPAC DPCs). The average size of the working database (IDTFLDB) is 30TB. It is so large because it has to keep data from a long period of time. Fig. 4 is a more realistic representation of the previous Fig. 3. The names of the systems are not relevant. It intends to show the amount of dependencies and tasks involved: there is not a sequential flow but many tasks are interleaved.

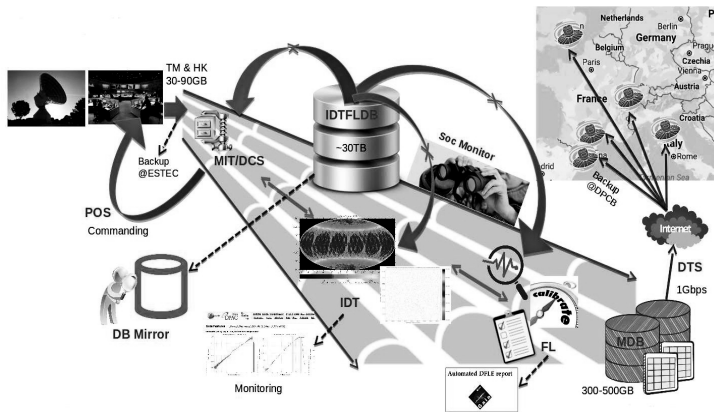


Figure 4. Complex SOC Daily Pipeline

3. The search for Robustness

Well before the Gaia launch the SOC team started the integration and validation of the software systems within the IT infrastructure. The focus was set on achieving a robust pipeline (highly reliable, scalable and available) capable of coping with error conditions various kinds. In order to achieve that goal a large suite of validation tests were conducted (system, integration, end-to-end tests) including Operational Rehearsals.

A dedicated DPAC Coordination Unit was in charge of providing simulated data at different levels (crucial to get realistic tests and meaningful outcomes) and through formal Test Readiness Reviews and Test Review Boards the quality of the test campaigns were guaranteed and the outputs traced. In parallel, additional software engineering and preparatory operational activities were done, e.g., organizing Configuration Control Boards, defining workflows and interfaces, identifying stakeholders, creating detailed procedures, etc. resulting in a seemingly robust, integrated pipeline.

However, quickly after launch it turned out that the systems were by far not as robust as we had expected. Numerous software bugs, unexpected loops and new de-

dependencies turned up almost daily and were difficult to deal with efficiently in the heat of real operations and the pressure to keep up with the non-stop flow of incoming telemetry data (to avoid backlogs). This proved quickly that the level of system robustness aimed at before launch was difficult to reach by conventional means. Fig. 5 shows a typical processing flow diagram from these early mission days. Disregarding the exact meaning of the coloured curves, the diagram illustrates gaps and delays (when the coloured symbols are below the diagonal dashed line) resulting from the aforementioned problems. All this was impossible to anticipate before launch.

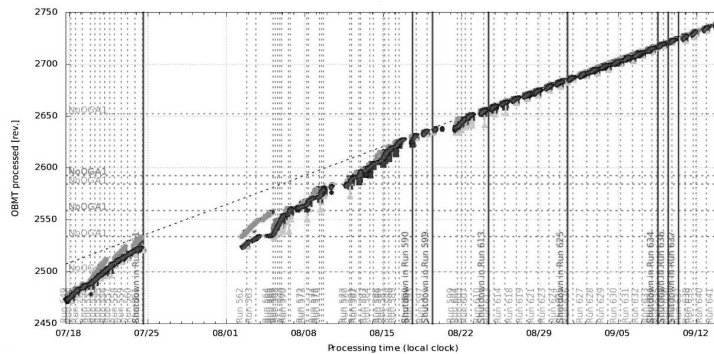


Figure 5. Processing flow diagram showing constant delays and gaps

4. A step further: The search for Antifragility through DevOps

Realizing that a robust system should resemble a castle, designed to fully protect the inside and repel any attack from the outside, it became clear that in order to fulfill the demanding requirements of the daily pipeline it was needed to get an integrated system capable of being flexible and adaptable: We call this antifragile.

The concept of “Antifragility” involves being able to stress the system in a way that it provides continuous feedback that can be incorporated fast. In this way, the effort is not on repelling changes and problems (as it is not possible to foresee all of them) but assimilating them to be stronger. And velocity is the key factor to achieve it.

DevOps appears in this context as the way to get antifragile operations. By embracing DevOps practices the pipeline processing became smooth even in cases of contingencies and the operations team increased their effectiveness:

- The processing of the daily pipeline became more stable over longer periods of time. Fig. 6 is a typical processing flow nowadays. The system is better able to keep up with the requirements on performance and data completeness (they are fulfilled practically at any time).
- With faster procedures, freed of “waste” (= what does not add any value) and more automated tools, the downtimes were reduced (in number and duration).
- More realistic tests allowed us to constantly challenge the ability to detect and recover from failures.

- By having more time to spend on improving the system rather than “fire fighting” new ways to do the same with less are being found (implying saving costs, e.g. reducing the number of HW resources).
- Automatically gathered metrics guarantees the quality of the data produced.
- The team remains motivated, avoiding burn-outs by a good prioritization of the tasks and facing new challenges even though the system is consolidated.

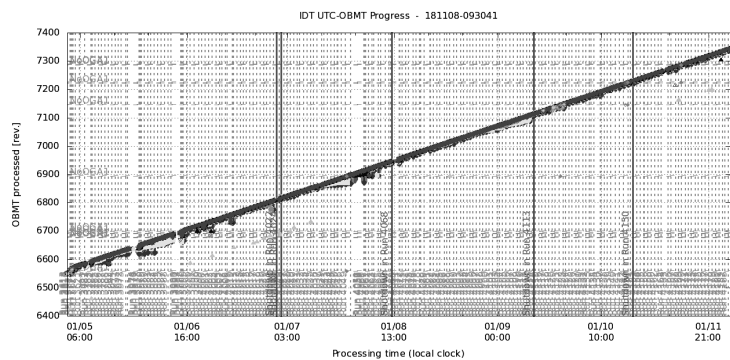


Figure 6. Typical current processing flow of the Daily Pipeline.

DevOps is often associated only with the usage of new tools and technologies as e.g. containers or micro-services. But DevOps is not just a set of tools brought together to do some magic. It requires a cultural change where the concept of automation, gathering metrics, lean (reducing waste) and collaboration are prioritized and always pursued. Obviously, without proper tools the collection of metrics or the processes automation become complicated tasks. Fig. 7 shows the technology stack used in the development and operations of the SOC Daily Pipeline. Some of them are very consolidated and key like Jenkins and some others (as e.g. Splunk) are currently being introduced.

It might be arguable whether the said improvements arrived naturally just because



Figure 7. SOC Daily Pipeline Technology Stack

with time the systems were gradually fixed and freed from software bugs and not because a new culture of fostering automation and collaboration were promoted. However, although the former is certainly true, embracing new practices have brought tangible benefits: systems have still bugs but there are mechanisms to find them fast causing only minimal disruptions. Manual intervention is less and less needed and the procedures are better tested so that operations are more reliable and the risks decreased. The following sections describe the main achievements over the last years.

5. Continuous Integration, Delivery, Artifact Repository and Infrastructure

Jenkins (Fig. 8) is the main Continuous Integration tool and a major contributor to the enhanced system’s stability. Jenkins pipelines have been recently introduced for the implementation and integration of complex jobs, encompassing building, testing, quality assurance and deployment, a big step towards automation. SonarQube is the source analysis tool for continuous inspection of code quality that is fully integrated with Jenkins. It provides a centralized place to share these QA metrics. Fig. 9 is a snapshot showing the type of information provided. For storage of software artifacts Nexus Professional Repository Manager is the tool used. The IT infrastructure is managed by the Science IT team at ESAC (SITU) that is also focusing efforts on automating most of their tasks. They use Puppet and Ansible for deploying, patching and configuring the items under their responsibility.

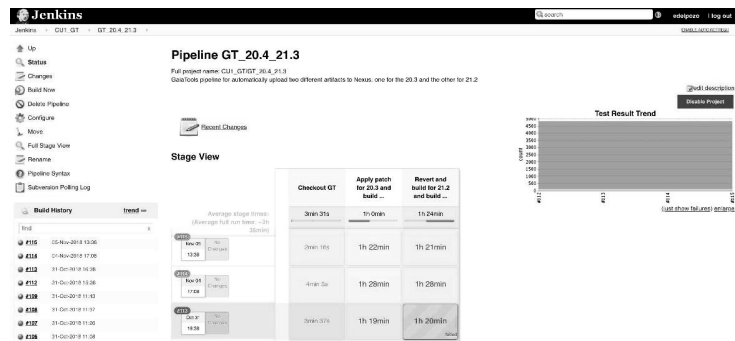


Figure 8. Jenkins usage in Gaia SOC

6. Testing Activities

The integration testing activities have always faced a main challenge, that is, the difficulties of reproducing realistic operational scenarios in a test environment. The main constraint was dealing with the volume of the working Intersystems Cache database (30 TB size on average). In the past it was very difficult and extremely time consuming to conduct one test in conditions similar to operations (let alone repeated tests). A major breakthrough was made applying NetApp (our storage vendor) technologies to clone the operational database any time with no impact in operations. It is now possible to do small, simple, fast and completely representative experiments and trials. Moreover, in

- Limit work in progress. This is the limit for how many items each operator will work on at the same time. It is key for avoiding the time wasted every day in switching from one activity to the other and helps on finishing activities that otherwise would take much longer having many other to address at the same time.
- Manage the work flow. It is the team that prioritizes the tasks such that an excessive amount of tasks (planned or unexpected) overload the team.

The operations team members gather every morning. The physical whiteboard and the sticky notes are being replaced by a digital board in Jira. The main benefit of using Jira is the metrics provided, very much useful to identify trends that might imply bottlenecks or other problems. Since the adoption of Kanban the throughput of issues resolution has increased considerably.

The team invented a role (that rotates every week) known as “Person of Interest” (POI) in charge of resolving all the issues from the other team members. The POI reviews the procedures followed and the information provided, suggesting changes when things are not clear enough. This is an excellent way to keep all the team trained and aware of all the activities ongoing and also an additional way to improve the procedures and processes followed. Fig. 11 shows an example of the reporting obtained by Jira from the Kanban board.

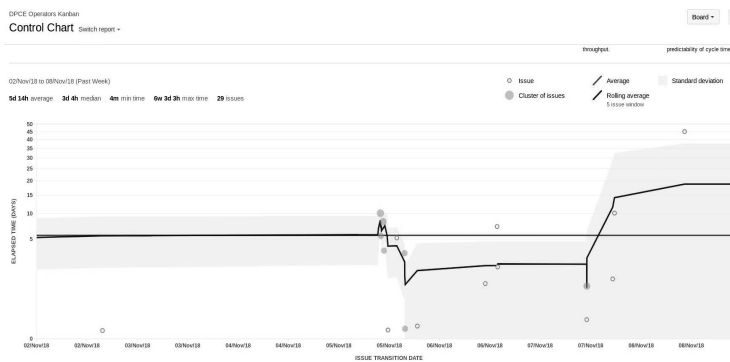


Figure 11. Control Chart of the Kanban board

8. Procedures and Runbooks

The procedures and runbooks (understood as groups of procedures) are being moved from a static documentation to Jupyter Notebooks. Markdown text, executable code and output all inside a single document fit the procedures needs very well. The inter-activity avoids them becoming obsolete as they are often executed and reviewed. In addition, automating the procedure steps decreases the risks of manual interventions. The procedures report automatically into the operations logbook (eLog) saving unnecessary time to the operator. Fig. 12 is an example of a procedure implemented as a Jupyter document.

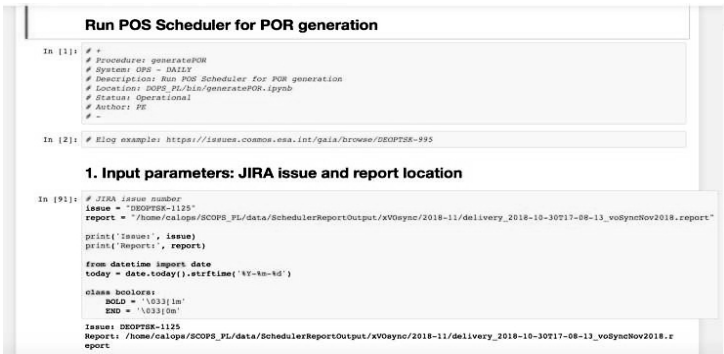


Figure 12. Procedure implemented in Jupyter Notebook - Python 3

9. Conclusions

By embracing some DevOps practices the Gaia SOC enhanced drastically the stability of the pipeline processing and the effectiveness of the team. Automation, avoiding unnecessary tasks (lean), collecting metrics to get knowledge and collaboration are the fundamental drivers of the change. This is a continuous learning process though, that can’t stop. The goal is to collect feedback of the behaviours of the processes and the team that can be used for improving them and achieving antifragile systems.

References

Hernandez, J., & Hutton, A. 2015, in Astronomical Data Analysis Software and Systems XXIV, edited by A. Taylor, & E. Rosolowsky, vol. 495 of Astronomical Society of the Pacific Conference Series, 47

O’Mullane, W. 2014, in Astronomical Data Analysis Software and Systems XXIII, edited by N. Manset, & P. Forshay, vol. 485 of Astronomical Society of the Pacific Conference Series, 487

Siddiqui, H. 2014, in Proceedings of the SPIE, vol. 9149 of SPIE, 9

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Agile and DevOps from the Trenches at ASTRON

G. M. Loose

ASTRON, Dwingeloo, The Netherlands; loose@astron.nl

Abstract. A few years ago the software development teams at ASTRON decided to adopt the Agile/Scrum software development method. We are building instruments and software that push technological boundaries. Requirements often lack sufficient detail and are subject to constant change, whilst the first data from a new instrument or early prototype become available. The unknown unknowns largely outnumber the known unknowns. Agile/Scrum has proven to be successful in situations like these.

We stumbled and fell, but gained a lot of experience in how Agile development techniques can be used in the scientific arena. We learned what works, and what does not work. We became more and more convinced that Agile/Scrum can be very effective in the area of Scientific Software development. In this presentation I would like to take you by the hand and revisit the journey we have made, in the hope that you will learn from the mistakes that we have made, and the lessons that we have learned.

1. How it all started

Late 2011, software development for the LOFAR telescope was in crisis. In the years before, focus had been on getting the instrument to work, taking short-cuts where needed. Little time was spent in making the instrument ready for operations. After the official opening in 2010, pressure on the software team to deliver new features and processing pipelines increased. However, many of the feature requests lacked clear requirements. Moreover, the huge technical debt created in the previous years brought development to a crawl. How did we end up in this situation?

1.1. From Waterfall ...

Much of the early software development was done in a Waterfall-like way. Waterfall works great if you exactly know what you need to build; if all the requirements are known and understood, and if you exactly know what technology you are going to use. In a scientific environment, this is usually not the case. Requirements (both user- and system requirements) constantly change, and you often operate at the boundary of what is technically feasible. Projects are generally complicated or complex. This is a regime where Agile is known to work very well.

1.2. ... To Agile

What makes Agile different from Waterfall? Superficially, Agile is just Waterfall using (very) short development cycles, of weeks instead of years. After each cycle, you redo all the Waterfall-steps. This way, you can adapt to change much more rapidly. But Agile is more than that. It is built on four principles described in the *Manifesto for*

Agile Software Development.¹ Scrum is merely a framework that aims to implement these principles.

2. Scrum in practice

Going Agile/Scrum is easier said than done. The concepts are easy to understand, but difficult to master. Being new to Agile/Scrum, working in a project-oriented organization, we soon fell in our first pitfall.

2.1. Product Owner and Sprint Reviews

We underestimated the role of the Product Owner. It is by no means obvious that a project-oriented organization should have someone who is responsible for a product; a product whose lifetime generally exceeds that of the project by far. Only recently have we, as an organization, recognized the importance of a product life-cycle.

Not having a Product Owner is really problematic, even if you have involved users. It was one of the main reasons we did not have Sprint Reviews. And though we sometimes gave demos, we did not give them on a regular basis. As a result, the Development Team did not get essential feedback.

Lacking a Product Owner also led to poorly defined User Stories. By now, we know that it is hard to overestimate the importance of the Product Owner; he is essential for successful Scrum.

2.2. Sprint planning

One of the hardest things to do right is planning. Even after years of experience, we often find ourselves confronted with unfinished work at the end of a sprint. We have made extensive analyses to the root causes of these overruns. Some of the most common errors are:

1. Issues are not worked out in enough detail. As a result, part of the work is not identified during planning.
2. There is a tendency to make issues too broad. Limiting the scope of an issue helps to better estimate the required effort.
3. Effort is put in fixing things that are not part of original issue. It would be better to create a separate bug-fix ticket for this in the issue tracker.

But organizational issues also play a role. We have seen frequent changes in the composition of teams, when management decided to shift focus and priorities. Teams are created for the duration of a project, and disbanded at the end of a project.

2.3. How to handle architecture

We found it quite hard to do proper architectural design within sprints. It is like building a house where you lay the foundation piece by piece as you build each room. We identified a number of solutions to this problem.

One solution is to define a few preparatory sprints to lay the foundation of your system. It is important to limit the number of sprints used for this. Keep it time-boxed,

¹<http://agilemanifesto.org/>

or you run the risk to build features that you think might be useful in the future. Chances are nine out of ten that these features will not be used.

Another option is to consider the Product Owner a stake holder as well. This works quite well, because the main interest of the PO is to continuously improve the product he fosters.

Yet another idea is to consider a (sub)system as a user or stakeholder too. This is in some way similar to UML Use Case diagram where the sticky figure does not necessarily denote a user of flesh and blood, but can also be a (sub)system.

The biggest challenge is to design an architecture that is open for change, while at the same time avoid to over-generalize.

2.4. Team size

Do not make your teams too large. The ideal size depends a bit on the maturity of the team, but with more than nine people overhead starts to become problematic.

2.5. Bug-fixing versus feature development

During commissioning, a lot of issues will surface that need to be resolved quickly. This poses a serious risk that feature development slows down too much. The battle between bug-fixing and feature development can be approached in a number of ways.

One approach is to plan bug fixes as normal sprint tasks. Though this will result in slower response times, it should be the preferred approach. If there are many serious bugs, you may consider to allocate part of the available time, say 30%, for immediate bug-fixing, using a more Kanban-like approach, e.g., Scrumban. Alternatively, the PO could decide to dedicate a whole sprint to bug-fixing only.

3. What about DevOps?

3.1. Continuous Integration

Testing has always been an important part in our development chain. Most of the tests are at the functional level; some core components also have unit tests; and there are some integration, client/server-like, tests. We use Jenkins to do regular automated builds, and run these tests. Our test system consists of a scaled-down version of the actual instrument. It contains components that are identical to those used in the production system. Wherever possible, we try to limit the use of mocks and stubs.

3.2. Continuous Delivery

The APERTIF's Long-Term Archive (ALTA) project was the first to really use DevOps from the start. ALTA is built completely from scratch, starting with a virtual machine that is created using Vagrant, a virtualization management tool. System configuration and software deployment is done using Ansible, a provisioning automation tool. This way, we can create a portable, disposable, consistent, and testable environment, which can be made part of the code base, and used for automating build, test, and deployment.

3.3. Containerization

Recent efforts to bring LOFAR post-processing pipelines to the user led to the use of containerization. A common problem with many of these pipelines is that they depend

on different, often conflicting versions of external software packages or libraries. Besides, some pipelines are still immature, and cause stability problems for the whole processing system. Docker to the rescue.

4. Lessons learned

4.1. What worked for us

Planning has improved overall. We have much more grip on the progress being made. We can make an accurate planning for the next milestone, and good ball-park estimates for future milestones.

Software quality has improved. We now have a reasonably stable trunk, thanks to the use of feature branches. And the team has much more focus, thanks to the short development cycles.

Last but not least, we have much more involvement of users and commissioners.

4.2. What did not work for us

We still do not really work as a Scrum team. Team members are often too specialized in what they do, which makes it hard to take over someone else's work.

Occasionally, we encounter situations where there are simply too many unknowns, or we face unexpected setbacks.

4.3. What we found hard

Planning for the unknown is one of the hardest things to do. We found that the best way forward is to define a Spike, a special type of time-boxed Story that is used to drive out risk or uncertainty.

Another big challenge is to properly handle software architecture, as is described in section 2.3

4.4. Improved understanding means improved planning

One of the most important lessons learned is that the quality of the planning improves significantly once the problem to be solved is better understood. This can be summarized in a number of dos and don'ts:

- Do not start to work on stories that are unclear
- If stories are too big, chop them up
- Break-down each story into smaller tasks
- Involve all stakeholders; Operators and Science Support are often forgotten
- ... And make sure you have a Product Owner

5. Conclusion

Agile/Scrum works! But it requires *organizational* change, *social* change, and a team that is willing to *continuously improve* itself. This is *not* a technical challenge, but a *social* challenge!

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Fundamentals of Effective Cloud Management for the New NASA Astrophysics Data System

Sergi Blanco-Cuaresma, Alberto Accomazzi, Michael J. Kurtz,
Edwin Henneken, Carolyn S. Grant, Donna M. Thompson, Roman Chyla,
Stephen McDonald, Golnaz Shapurian, Timothy W. Hostetler,
Matthew R. Templeton, Kelly E. Lockhart, Kris Bukovi, and Nathan Rapport

*Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge,
MA 02138, USA sblancocuaresma@cfa.harvard.edu*

Abstract. The new NASA Astrophysics Data System (ADS) is designed with a service-oriented architecture (SOA) that consists of multiple customized Apache Solr search engine instances plus a collection of microservices, containerized using Docker, and deployed in Amazon Web Services (AWS). For complex systems, like the ADS, this loosely coupled architecture can lead to a more scalable, reliable and resilient system if some fundamental questions are addressed. After having experimented with different AWS environments and deployment methods, we decided in December 2017 to go with Kubernetes as our container orchestration. Defining the best strategy to properly setup Kubernetes has shown to be challenging: automatic scaling services and load balancing traffic can lead to errors whose origin is difficult to identify, monitoring and logging the activity that happens across multiple layers for a single request needs to be carefully addressed, and the best workflow for a Continuous Integration and Delivery (CI/CD) system is not self-evident. We present here how we tackle these challenges and our plans for the future.

1. Introduction

The NASA Astrophysics Data System (ADS; Kurtz et al. 2000) is a key bibliographic service for astronomical research. ADS content has steadily increased since its early years (Grant et al. 2000), containing now more than 13 million records and 100 million citations including software and data citations (Accomazzi 2015). After several iterations, its original architecture (Accomazzi et al. 2000) and user interface (Eichhorn et al. 2000) have evolved to address growing maintenance challenges and to adopt newer technologies that allow more advanced functionality (Chyla et al. 2015; Accomazzi et al. 2015, 2018).

The new ADS is designed with a service-oriented architecture (SOA), containerized using Docker¹, orchestrated by Kubernetes² and deployed in Amazon Web Services³ (AWS). We have been using this platform for almost a year now, both in our

¹<https://www.docker.com/>

²<https://kubernetes.io/>

³<https://aws.amazon.com/>

2.1. Monitoring

Making sure the whole system is healthy and responding to users' requests is a priority. We developed a custom monitoring tool that emulates users' behavior (e.g., executing searches, accessing libraries, exporting records, filtering results) and alerts us to unexpected results or errors via Slack⁵. This emulation happens with a high cadence of the order of several minutes. Historical data is also accumulated and daily reports are generated to measure trends and improvements that could be correlated with microservices updates or infrastructure changes.

2.2. Logging

Responding to a single user request may involve multiple microservices (e.g., libraries, Solr search service) and different data requests (e.g., bibcodes in a library, records in Solr). At the very first step, when the user request reaches the AWS application load balancer, a trace identifier is attached to the HTTP request and we propagate it for each required internal request inside our infrastructure. All the microservices output logs to stdout, including key information such as the trace identifier and the user's account identifier. Logs are captured by Fluent Bit⁶ and distributed to Graylog⁷ and AWS CloudWatch via Fluentd⁸.

2.3. Deploying

The deployment of new microservice releases is automatically managed by Keel⁹. The developers push new commits to GitHub¹⁰ and/or make releases, which triggers unit testing via Travis¹¹ continuous integration and image building via Docker hub¹². When a new image is built, Keel deploys it directly to our development environment (each pushed commit) or to our quality assurance environment (each new release). Confirmation to deploy a release in production is provided via Slack, where Keel reports its operations and reacts to developers' approvals.

3. Future plans

Several microservices still require manual intervention in order to deploy new releases, Keel does not cover all our development cases and we are working on a new custom tool to meet our needs (after having discarded other tools available in the market due to their complexity). We seek to fully automate the deployment process, while ensuring

⁵<https://slack.com/>

⁶<https://fluentbit.io/>

⁷<https://www.graylog.org/>

⁸<https://www.fluentd.org/>

⁹<https://keel.sh/>

¹⁰<https://github.com/>

¹¹<https://travis-ci.org/>

¹²<https://hub.docker.com/>

traceability and easy roll-backs based on automatic functional tests from our monitoring tool. Additionally, to reduce the required resources and simplify operations, we will evaluate other engines for searching through our logs such as Kibana via Elastic-Search¹³ (provided by AWS).

References

- Abbott, T. M. C., Abdalla, F. B., Allam, S., Amara, A., Annis, J., Asorey, J., Avila, S., Ballester, O., & et al. 2018, arXiv e-prints. 1801.03181
- Accomazzi, A. 2015, in Science Operations 2015: Science Data Management - An ESO/ESA Workshop, 3
- Accomazzi, A., Eichhorn, G., Kurtz, M. J., Grant, C. S., & Murray, S. S. 2000, *Astronomy and Astrophysics Supplement Series*, 143, 85. astro-ph/0002105
- Accomazzi, A., Kurtz, M. J., Henneken, E., Grant, C. S., Thompson, D. M., Chyla, R., McDonald, S., Shaulis, T. J., Blanco-Cuaresma, S., Shapurian, G., Hostetler, T. W., Templeton, M. R., & Lockhart, K. E. 2018, in American Astronomical Society Meeting Abstracts #231, vol. 231 of American Astronomical Society Meeting Abstracts, 362.17
- Accomazzi, A., Kurtz, M. J., Henneken, E. A., Chyla, R., Luker, J., Grant, C. S., Thompson, D. M., Holachek, A., Dave, R., & Murray, S. S. 2015, in Open Science at the Frontiers of Librarianship, edited by A. Holl, S. Lesteven, D. Dietrich, & A. Gasperini, vol. 492, 189
- Araya, M., Osorio, M., Díaz, M., Ponce, C., Villanueva, M., Valenzuela, C., & Solar, M. 2018, *Astronomy and Computing*, 25, 110
- Chyla, R., Accomazzi, A., Holachek, A., Grant, C. S., Elliott, J., Henneken, E. A., Thompson, D. M., Kurtz, M. J., Murray, S. S., & Sudilovsky, V. 2015, in *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*, edited by A. R. Taylor, & E. Rosolowsky, vol. 495, 401
- Eichhorn, G., Kurtz, M. J., Accomazzi, A., Grant, C. S., & Murray, S. S. 2000, *Astronomy and Astrophysics Supplement Series*, 143, 61. astro-ph/0002102
- Farias, H. A., Ortiz, D., Núñez, C., Solar, M., & Bugueno, M. 2018, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 10707, 107072R
- Grant, C. S., Accomazzi, A., Eichhorn, G., Kurtz, M. J., & Murray, S. S. 2000, *Astronomy and Astrophysics Supplement Series*, 143, 111. astro-ph/0002103
- Kurtz, M. J., Eichhorn, G., Accomazzi, A., Grant, C. S., Murray, S. S., & Watson, J. M. 2000, *Astronomy and Astrophysics Supplement Series*, 143, 41. astro-ph/0002104

¹³<https://www.elastic.co/products/kibana>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Versioned Executable User Documentation for In-development Science Tools

C. Boisson¹, J. E. Ruiz², C. Deil,³ A. Donath,³ and B. Khelifi⁴

¹*LUTH, Observatoire de Paris, Paris, France; catherine.boisson@obspm.fr*

²*Instituto de Astrofísica de Andalucía - CSIC, Granada, Spain; jer@iaa.es*

³*Max-Planck-Institut für Kernphysik, Heidelberg, Germany*

⁴*APC - AstroParticule et Cosmologie, Université Paris Diderot, Paris, France*

Abstract. One key aspect of software development is feedback from users. This community is not always aware of the modifications made in the code base, neither they use the tools and practices followed by the developers to deal with a non-stable software in continuous evolution. The open-source Python package for gamma-ray astronomy Gammapy, provides its user community with versioned computing environments and executable documentation, in the form of Jupyter notebooks and virtual environment technologies that are versioned coupled with the code base. We find that this set-up greatly improves the user experience for a software in prototyping phase, as well as provides a good workflow to maintain an up-to-date documentation.

1. The Gammapy package

Gammapy¹ (Deil et al. 2017) is a community-developed, open-source Python package for high-level γ -ray data analysis, built on Numpy (Oliphant 2006) and Astropy (Greenfield et al. 2013). It provides functionalities to create sky images, spectra, light curves and source catalogs from event lists and instrument response information, determining the position, morphology and flux of γ -ray sources. Gammapy is a prototype for the Cherenkov Telescope Array (CTA) science tools. It has been used to simulate and analyze data for the CTA, as well as for the main IACT (Imaging Air Cherenkov Telescopes) facilities like H.E.S.S., MAGIC, VERITAS, and the Fermi-LAT telescope.

2. The user role for *in-development* software

The Gammapy package is a software in evolution, rapidly changing, where many developments are ongoing. As such, it is a place for γ -ray astronomers to share their code and contribute. So far, developer coding sprints have been scheduled to happen with the same regularity of user hands-on sessions held in CTA consortium meetings. A small collection of tutorials, in the form of Jupyter notebooks (Kluyver et al. 2016), was originally provided for learning purposes as a complement to the web published

¹<https://gammapy.org>

documentation. The documentation and notebooks had to be updated frequently and separately. With the rise in the contributor and user base, as well as in the development activity, it emerged the need to have the notebooks integrated in the documentation, and versioned coupled with the code base. Users should easily bring to life the different versions of the published tutorials, so they could be able reproduce and re-use them for their own goals.

3. Integrating versioned and executable user documentation

A graphic description of the technical set-up may be found in Figure 1. The source files for the documentation, the Jupyter notebooks, and the code base are maintained in a single GitHub repository². The code base and notebooks are continuously built and tested in different environments using the continuous integration service Travis CI.

The notebooks are stored stripped of their output cells, which greatly helps in identifying differences in the contribution review. We have implemented our own solution for execution validation of the notebooks: these are executed with *jupyter nbconvert*, the output cells parsed and, in case an error is found in one of the output cells, validation fails. Code formatting of their input cells is also done parsing their content with the help of the *Black* package.

During the documentation building process, the stripped notebooks are executed, so to refill their output cells, and later merged into the documentation with Sphinx and the *nbsphinx* extension. It is possible to skip the notebooks integration, since the execution step may be quite time consuming. In this sense, only fast simple notebooks are chosen to be part of the tutorials. The resulting web published tutorials³ provide links to versioned *playground* spaces in myBinder (Project Jupyter et al. 2018), where they may be executed on-line in virtual environments hosted in the myBinder infrastructure. These environments are built with the help of a *Dockerfile* placed at the top level of the GitHub repository.

The user may retrieve specific *tutorial bundles* using the *gammapy* download command. One *tutorial bundle* is composed of a *conda* file environment, Jupyter notebooks, and the datasets needed to reproduce them. For each of these bundles we specify, in centralized index lookup files, the required computing environment, which tutorials and datasets to provide, and where to fetch them from. Deterministic computing environments are defined in the form of *conda* configuration files, with pinned version numbers for each dependency package.

4. Command line tools and functionalities

We have developed a set of tools for users to download the *tutorial bundles*, and for documentation maintainers to prepare and validate them. The *gammapy* download command shown in Figure 2, provides users with the means to retrieve a *tutorial bundle* or any tutorial-related asset (i.e. dataset or Jupyter notebook) for a specific version of the Gammapy package, so they can activate and use it at their will.

²<https://github.com/gammapy/gammapy>

³<https://docs.gammapy.org/0.8/tutorials.html>

On the other side, the `gammapy jupyter` command provides documentation maintainers with a tool to seamlessly manage and integrate Jupyter notebooks into the documentation. This command provides functionalities for execution validation, code formatting, output cells stripping and straightforward execution of the notebooks.

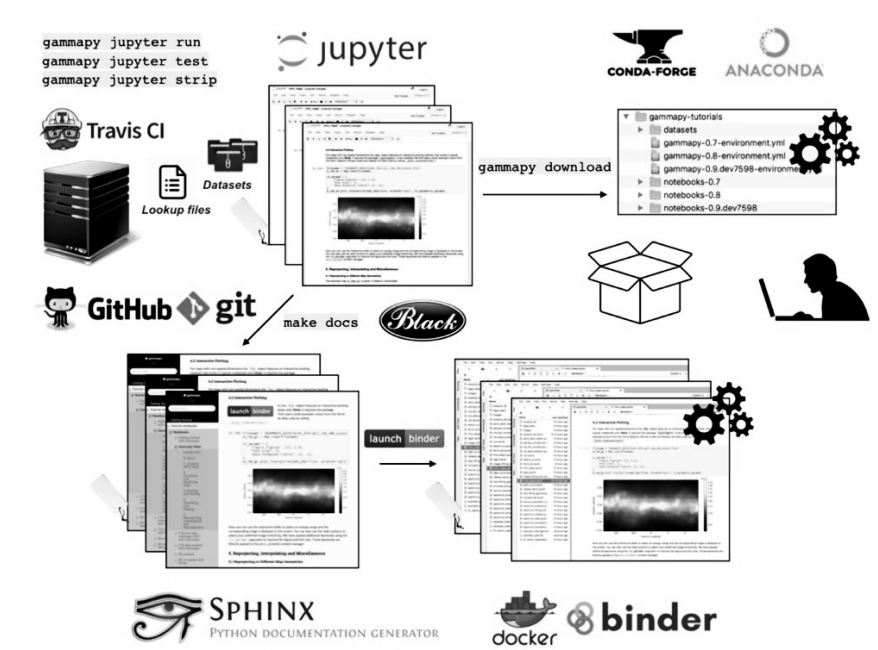


Figure 1. Technical set-up for building and shipping *tutorial bundles*. The maintainers build the documentation from stripped Jupyter notebooks using Sphinx. The resulting published tutorials provide links to access myBinder *playground* spaces, where the notebooks may be executed on-line. They may be also downloaded to the user desktop, together with the *conda* virtual environment and the datasets needed.

```
$ gammapy download tutorials --release 0.8
INFO:gammapy.scripts.downloadclass:Content will be downloaded in gammapy-tutorials/notebooks-0.8
Downloading files [=====] 100%
INFO:gammapy.scripts.downloadclass:Content will be downloaded in gammapy-tutorials/datasets
Downloading files [=====] 100%

**** Enter the following commands below to get started with Gammapy
cd gammapy-tutorials
conda env create -f gammapy-0.8-environment.yml
conda activate gammapy-0.8
export GAMMAPY_DATA=/Users/jer/Desktop/gammapy-tutorials/datasets
jupyter lab
```

Figure 2. Users may retrieve versioned *tutorial bundles*, or specific related digital assets like notebooks and datasets, using the `gammapy download` command.

5. Conclusions

We have presented a novel set-up to publish and distribute executable documentation version coupled with the code base of the Gammapy package. These *tutorial bundles* are composed of a *conda* virtual environment, Jupyter notebooks, and the datasets needed to reproduce them. The web published tutorials may be also executed on-line for learning purposes in myBinder *playground* spaces. We find that this set-up greatly improves the user experience for a software in prototyping phase, where maintenance and delivery of the *tutorial bundles* are done with on purpose command line tools.

Acknowledgments. The authors would like to thank the following projects and services that are used in the exposed work: *Git*⁴, *GitHub*⁵, *Travis CI*⁶, *Anaconda*⁷, *conda*⁸, *conda-forge*⁹, *Sphinx*¹⁰, *nbsphinx*¹¹, and *Black*¹². This work was partially funded by ASTERICS (<http://www.asterics2020.eu/>), a project supported by the European Commission Framework Programme Horizon 2020 Research and Innovation action under grant agreement n. 653477

References

- Deil, C., Donath, A., Owen, E., Terrier, R., Bühler, R., & Armstrong, T. 2017, Gammapy: Python toolbox for gamma-ray astronomy, Astrophysics Source Code Library. 1711.014
- Greenfield, P., Robitaille, T., Tollerud, E., Aldcroft, T., Barbary, K., Barrett, P., Bray, E., Crighton, N., Conley, A., Conseil, S., Davis, M., Deil, C., Dencheva, N., Droettboom, M., Ferguson, H., Ginsburg, A., Grollier, F., Moritz Günther, H., Hanley, C., Hsu, J. C., Kerzendorf, W., Kramer, R., Lian Lim, P., Muna, D., Nair, P., Price-Whelan, A., Shiga, D., Singer, L., Taylor, J., Turner, J., Woillez, J., & Zabalza, V. 2013, Astropy: Community Python library for astronomy, Astrophysics Source Code Library. 1304.002
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. 2016, in Positioning and Power in Academic Publishing: Players, Agents and Agendas, edited by F. Loizides, & B. Schmidt (IOS Press), 87
- Oliphant, T. 2006, Guide to NumPy (Trelgol Publishing)
- Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, Pacer, M., Yuvi Panda, Fernando Perez, Benjamin Ragan Kelley, & Carol Willing 2018, in Proceedings of the 17th Python in Science Conference, edited by Fatih Akici, David Lippa, Dillon Niederhut, & M. Pacer, 113

⁴<https://git-scm.com>

⁵<https://github.com>

⁶<https://travis-ci.com>

⁷<https://www.anaconda.com>

⁸<https://conda.io>

⁹<https://conda-forge.org>

¹⁰<https://www.sphinx-doc.org>

¹¹<https://github.com/spatialaudio/nbsphinx>

¹²<https://github.com/ambv/black>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Development, Tests, and Deployment of Web Application in DACE

Julien Burnier, Fabien Alesina, and Nicolas Buchschacher

University of Geneva, Geneva, Geneva, Switzerland; julien.burnier@unige.ch

Abstract. The Data and Analysis Center for Exoplanets (DACE) is a web platform based at the University of Geneva (CH) dedicated to extrasolar planets data visualisation, exchange and analysis. This platform is based on web technologies using common programming languages like HTML and Javascript for the front-end and a Java REST API for the back-end. Over the last 12 months, the process to maintain, develop, test and deploy the applications has been dramatically improved to facilitate the maintenance and the integration of new features. The goal of such automation is to let more time to focus on development and reduce the duplicated work. To achieve this result, we migrated our Java application to the Maven software project management and added unit tests. We implemented a pipeline on GitLab which consists of executing the tests and deploy the application in a dev environment at every commit. The front-end side is then tested using the Selenium web browser automation to simulate the user - website interactions and compare the new results with the old ones. Once all the tests are validated, a manual action on the GitLab interface can be done to deploy the application on the official web site and we ensure the compatibility of the new features with the production version. We are currently working to have a very complete set of tests on both back and front end in order to remove the manual part of production deployment and to have a fully automated integration of our applications.

1. Introduction

About one year ago, the DACE platform suffered deployment issues and stability. Sometimes, our team was afraid to go on production because we didn't deploy new code since 2-3 months. Also a quick bug fix to deploy was complicated because the code was not updated since a long time. To facilitate development and deployment we naturally decided to implement continuous integration.

2. Java code with maven and unit tests

The first thing to do was to encapsulate existing code to a project management tool which manage dependencies and simplify compilation, test and jar/war creation. We chose maven as it remains the reference on Java. Then we added unit test to existing code where possible. To do this we followed mvn convention and started using JUnit. Finally, the idea for backend API for example, was to apply Test Driven Development when possible.

3. GitLab – Continuous Integration tool

Then we needed a tool to do continuous integration. Fortunately, other project inside our university started to use GitLab. We migrated our code from svn to git and pushed code inside GitLab. After that, a gitlab-runner must be installed where you want to run your build. And finally, we added a *.gitlab-ci* file into every project. You can see an example of our GitLab file (only with stage Test, Build and staging deployment) on Figure 1. With this small configuration file you already have tests runned after each

```

stages:
  - build
  - staging
  - release
  - production

Maven Test and Build:
stage: build
script: mvn clean install
# Need this config to pass the jar to next stages
artifacts:
  paths:|
    - target/*.war
    expire_in: 1 day

Deploy to Staging:
stage: staging
environment:
  name: Staging
script:
  - export WEBAPP_FULL_NAME=$(basename target/*.war)
  - cp target/$WEBAPP_FULL_NAME /opt/tomcat/dace-webapps
  # ln -fsn will do the following : -f overrides if symlink already exists. -n will not follow previous symlink.
  # -s : standard option to have a symlink and not a hard link
  - export WEBAPP_NAME=$(echo $WEBAPP_FULL_NAME | sed 's/-.*/')
  - export WEBAPP_NAME="${WEBAPP_NAME}.war"
  - ln -fsn /opt/tomcat/dace-webapps/$WEBAPP_FULL_NAME /opt/tomcat/webapps/$WEBAPP_NAME

```

Figure 1. Example Gitlab code

commit and you'll receive an email in case of failure. The test and build script is simply *mvn clean install* and maven handle the rest. For deployment on staging (which is dev on our architecture), we do a cp of the war/jar and create a symbolic link to easily rollback in case of failure. For frontend part, the idea is to deploy via rsync on servers and then use Selenium.

4. Selenium

Before going to production, web tests should be done to ensure having no bugs and no regression (same as unit test but on visual part). To do this we use selenium. This tool simulates user interaction on a website. To facilitate selenium test development, we adopted Page Object Pattern. The Page object pattern let you split your selenium test code on each web page. For example, you can do a test on index.html and then click on about.html and create a test *AboutPageTest.java* to test this specific page. Example of home page test on Figure 2

5. Deployment

The deployment is also handled by GitLab. It needs to be defined in *.gitlab-ci.yml*. On our side, after every commit, we deploy backend and frontend on dev environment. Then if everything is ok, the app is released deployed manually on one production


```

package ch.unige.dace.pages.home;

import ...

/**
 * Created by julien on 20.10.17.
 * <p>
 * Test the home page. Use FunctionalTest to setup selenium environment
 */
public class HomePagePolygonsTest extends FunctionalTest {

    @Test
    public void shouldDisplayHomePage() throws InterruptedException {

        Navigator.openDaceWebsite();

        HomePagePolygons homePagePolygons = new HomePagePolygons();

        assertThat(homePagePolygons.isDisplayed()).isTrue();

        List<LocatorAndPage> polygons = homePagePolygons.getPolygons();

        for (LocatorAndPage polygon : polygons) {
            Page page = Navigator.goToPage(polygon);
            if (Navigator.isPageOpenOnNewTab()) {
                Navigator.switchTab();
                assertThat(page.isDisplayed()).isTrue();
                Navigator.closeTab();
            } else {
                assertThat(page.isDisplayed()).isTrue();
                Navigator.goBack();
            }
        }
    }
}

```

Figure 2. Example Selenium test code

server and after some manual tests, deployed on the other production server. Architecture of DACE servers and deployment process can be seen on Figure 3

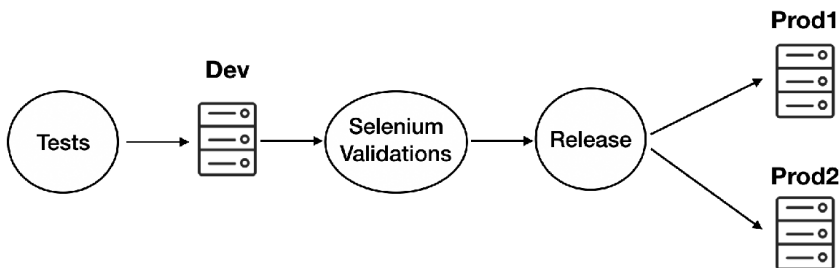


Figure 3. DACE continuous integration process

Acknowledgments. This work has been carried out within the framework of the National Centre for Competence in Research PlanetS supported by the Swiss National Science Foundation. The authors acknowledge financial support from the SNSF. This publication makes use of DACE, a Data Analysis Center for Exoplanets, a platform of the Swiss National Centre of Competence in Research (NCCR) PlanetS, based at the University of Geneva (CH).



Pascal Ballester entertained by the Geneva DACE team. (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

GDL – GNU Data Language 0.9.9

Alain Coulais,¹ Gilles Duvert,² Gregory Jung,³ Sylwester Arabas,⁴
 Sylvain Flinois,¹ and Amar Si Lounis¹

¹*LERMA (CNRS) & Observatoire de Paris, Paris, France;*
alain.coulais@obspm.fr

²*Univ. Grenoble Alpes, CNRS, IPAG, 38000 Grenoble, France*

³*Long Beach, California, USA*

⁴*Jagiellonian University, Cracow, Poland*

Abstract. On behalf of the GDL team, we announce the 0.9.9 release of GDL, the GNU Data Language. GDL is a free and open-source drop-in replacement for IDL, the Interactive Data Language, which has served the astronomical community for many years. This release comes 15 years after Marc Schellens first made GDL public, and it marks several milestones attesting to the maturity of the project. Among these are updates to the plotting functionalities, major advances in the Windows implementation, a native support for IDL save files, and performance improvements.

The presentation will summarize current availability of packaged versions of GDL, compatibility with popular IDL-written astronomy-related software, potential directions for the upcoming releases, and the key areas where GDL could benefit from further user feedback and new contributions.

1. Introduction

GDL is a *free/libre* clone of IDL, an interpreted language widely used in astronomy. GDL is redistributable under the terms of the GNU General Public License as published by the Free Software Foundation. Importantly, building and installing GDL on any computer (laptop, desktop, HPC) is a manageable task. Since we still have questions of that point, yes, GDL does interpret and run programs written in IDL syntax.

2. Move to GitHub

After more than 10 years of development hosting at SourceForge (thank you for all the years of hospitality!), GDL moved in April 2018 to GitHub¹. What did we expect? We hoped it would become easier for external users or contributors to make bugs reports and “pull requests,” i.e., to contribute. Clearly, it does! GDL development benefits now from the continuous integration (CI) workflow, performed using virtual machines

¹<https://github.com/gnudatalanguage/gdl>

deployed through Travis for Linux and OSX, and through Appveyor for MS Windows. The Travis builds test three major C++ compilers (gcc, clang & icc) in Linux and OSX environments with the scenarios : all options on, and most off. Another interesting feature of CI in GitHub is the CodeCov utility that provides an estimate of the test coverage ratio for the project.

3. Functionalities

Many incremental improvements in the code have been made (see NEWS file in the project). Although some aspects of IDL v8 are targeted (e.g., implied print, arrays defined by [...]), GDL attempts realistically to attain v7 syntax compatibility.

Major improvements were introduced in *LIST* and *HASH* functionalities. The *LIST* object class is (almost) completely compatible and is an early example of a built-in *GDL_OBJECT* (the functionality of the "legacy" version was limited). The */TOARRAY* option is fully operational for numeric types and for strings, and one can access list elements (read and write, including those in a *HASH* object) directly through indexing. Significantly, the *LAMBDA* facility is unavailable and so methods which would use that are not implemented. The *HASH* functionality is (almost) completely compatible. In-place direct array access is possible via indexing, including through elements that are *LIST* objects.

4. On compilation time

GDL is clearly slower than IDL on compilation of large codes in IDL syntax (~10 times slower). It was tested on various libraries and codes, including the Astro lib.², MPfit (Markwardt 2009), Coyote³ and Ulyss⁴ (Prugniel et al. 2011). A major slowdown related to intricate and cascading parenthesis (“[]” and “()”) was improved more than one year ago. If your code is “[]” vs “()” safe, you can run GDL with the option `--fussy` and benefit from some additional speedup. We believe that compilation speed is way less important than (1) the execution speed of basic operations (+-*/), mathematical primitives (COS/SIN, EXP ...) and the high level operations (INVERT, FFT, ...) and (2) validity and accuracy of the computations and functionalities.

5. Benchmarking

Some of the many functional improvements had consequences for benchmarks on basic operations. During summer 2018, serious checks, now available in the `benchmark/` sub-directory, revealed a need to improve the speed of basic operations. Now, most of them are in-line with IDL. We welcome feedback on any slowness. Multi-cores aspect was already developed in Coulais et al. (2014).

²<https://idlastro.gsfc.nasa.gov/>

³<http://www.idlcoyote.com/>

⁴<http://ulyss.univ-lyon1.fr>

6. Test suite and regression tests

The test suite in GDL has existed for years, extending year after year and critically helping us to avoid unwanted regression(s). Since the code has been hosted on GitHub, thanks to CodeCov, we know today that our test suite covers about 43% of the C++ codebase. Increasing the coverage is one of the highlights of the ongoing development work, however for some parts of the code, in particular the graphics and widgets subsystems, a robust workflow for automating tests is still to be figured out.

Manual tests are also based on running large codes, cross-checking results and time performances from IDL and GDL runs. When numerical outputs differ by a significant figure, it is time to check the code. In GDL, some parts of computation are done in Double but in Float in IDL, and can be more accurate.

As mentioned in the past, with the tests, we did find troubles in IDL too.

Please note that when deactivating dSFMT fast random generator (`--no-dSFMT`), for a given seed you will have the same random series in IDL & GDL, which can be useful for some numerical comparisons.

7. Packaging versus compilation

GDL has been packaged for years in most of the main distributions (see a up-to-date table in the `README.md` in GitHub) for Linux, OSX & *BSD. Warm thanks to the packagers. But the rate to deliver a release is now about one per year, and GDL should also benefit from compiling from scratch with a high level of optimization (`-O3`). GDL is compilable on most systems as long as they have sufficiently recent build tools available (`gcc ≥ 4.8` with OpenMP support, `CMake ≥ 3`). Users should be able to compile the current git version of GDL even in environments for which GDL has outdated packages or is not packaged yes. We do provide a limited support for such compilation.

The script `minimum_script4gdl.sh` in `scripts/` directory is a very simple way to compile a subset of GDL (some third party libs are not activated) on most POSIX compliant systems. It has been successfully tested on a lot of Linux versions (Debian, Ubuntu, CentOS, Fedora, ...) and all OSX flavors.

8. Relation with Debian astro project

Debian Astro⁵ contains also three packages well known by users of IDL: the Astro lib., the Coyote lib. and the MPfit lib. Since those libs. are just collection of routines in IDL/GDL syntax, they can be used on any system, and can be downloaded on any OS just by `wget`, `curl` or `git clone` commands. Then you can add their paths (with `PATH_ADD`) to run them in GDL (or in IDL).

We know the support of FITS format (compress or not; reading and writing) is working well since years with the Astro lib and GDL (last bug reported in 2013). Nevertheless we need feedback for any trouble with this lib. because its codes are still evolving, and we have no test suite now. The tests included in Coyote lib. run well as long as the file “`colors1.tbl`” can be located in `!dir`. The tests related to MPfit lib. may

⁵<https://blends.debian.org/astro/tasks/gdl>

differ from the results in IDL due to the random number generator. We know this lib. is largely used in GDL without problem as long as you know you may have numerical differences due to internal representations.

9. Why not a 1.0 version ?

Months ago, we anticipated delivery of a “1.0” version for this conference, but we descoped it, because of two blocking cases: (1) The quality and the coverage of the MS Windows version is not at the level of the Linux or OSX ones. Much of the difficulties are issues in the graphics subsystem. But linking with all the third party dependencies is still not simple. Help welcome. (2) The widgets are OK only on Linux or *BSD. They are not OK for OSX and MS Windows. It’s a pity because smart widgets like ATV⁶ are now available in GDL only on Linux.

10. User feedback and information

It is of high interest that end users give feedback to the GDL development team. Of course, active contribution is more than welcome. We think the move to GitHub will simplify the exchanges. Don’t hesitate to report bugs, if possible based on recent versions (0.9.9 or Git version). Users requests help us to make priorities. Since the apparition of *new graphics* in IDL 8.0, we received only one request for that. Don’t expect it soon ! Developing for GDL is a very good topic for internships. Students learn a lot and put online visible contributions. To receive sporadic announcements on GDL (few messages per year: releases, OS related tricks, new functionalities, critical bugs ...) it is a good idea to subscribe to the announce diffusion list⁷.

11. Conclusion

GDL is already a very stable version on Linux, OSX and *BSD systems, daily used by many end-users and also on HPC for large computations. You, end-users, are very welcome to give feedback, bug reports, request for functionalities on the GitHub depot. Obviously, pull requests are very welcome!

References

- Coulais, A., Schellens, M., Duvert, G., Park, J., Arabas, S., Erard, S., Roudier, G., Hivon, E., Mottet, S., Laurent, B., Pinter, M., Kasradze, N., & Ayad, M. 2014, in *Astronomical Data Analysis Software and Systems XXIII*, edited by N. Manset, & P. Forshay, vol. 485 of *Astronomical Society of the Pacific Conference Series*, 331
- Markwardt, C. B. 2009, in *Astronomical Data Analysis Software and Systems XVIII*, edited by D. A. Bohlender, D. Durand, & P. Dowler, vol. 411 of *Astronomical Society of the Pacific Conference Series*, 251. 0902.2850
- Prugniel, P., Vauglin, I., & Koleva, M. 2011, *A&A*, 531, A165. 1104.4952

⁶<https://www.physics.uci.edu/~barth/atv/>

⁷<https://sympa.obspm.fr/wvs/info/gdl-announces>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Automating Multimission Access: Rolling Out a Flexible Virtual Observatory-based Infrastructure

Theresa Dower¹ and Bernie Shiao²

¹*STScI, Baltimore, Maryland, USA; dower@stsci.edu*

²*STScI, Baltimore, Maryland, USA*

Abstract. As a part of The Mikulski Archive for Space Telescopes' (MAST) mission to develop and support user-friendly and scientifically useful search tools and foster inter-archive communication and interoperable standards, we integrate International Virtual Observatory (IVOA) standards in our infrastructure. These standard practices allow us to provide layers of simple and complex interfaces supporting both direct scientific research and inter-archive pipeline coordination. Here we discuss an architecture based on Table Access Protocol (TAP) services for MAST holdings, how we are automating the deployment and maintenance of these services to provide more data coverage with modern research software including Astropy, and how we can update these pieces individually to continue meeting unique technological challenges.

1. Introduction

As IVOA standards for data access have become more prevalent across astronomical archives, the functional lifetime of some standards has outstripped that of the technology stacks implementing them. With Simple Cone Search, Image Access, and Spectral Access services (referred to here as simple services) still being used alongside the new, more powerful but complex TAP standard (Dowler et al. 2011), we now have legacy services providing a simpler window into the same data, tied to a legacy technology stack, in parallel with TAP and stacks upon which TAP implementations depend. In rolling out software for a new standard providing a superset of existing functionality, MAST has an opportunity to create a new stack and move new and legacy projects together. Smart clients like the MAST Portal can use TAP directly. Replacement simple services can access TAP as an internal layer as they map directly to a subset of TAP functionality easily defined by canned queries. And MAST can leverage existing open TAP client and server software, contributing to it for its own infrastructure and for the wider community.

2. Current Status and System Design

To meet the technological challenge of heterogeneous data in collections of widely varying size, we chose to first create a new TAP stack on platforms already handling our largest catalogs, designed modularly with open-sourced tools where possible. Each TAP service released in the building phase focused on achieving the most data coverage first, then serving more complex catalogs. Currently available through TAP

are all MAST holdings in the Common Archive Observation Model (CAOM), The Hubble Source Catalog, a filtered product of Pan-STARRS 1 DR1, and a searchable IVOA Metadata Registry. The second phase, automating TAP service deployment for new catalogs and High-Level Science Products (HLSPs), is underway, with scripted schema metadata and spatial index generation complete and an automated testing suite in progress.

Remaining phases of effort begin with determining the priority of holdings to roll out coverage for in TAP services and new simple services as applicable, based on archive scientist input and usage logs. We will develop the thin simple service layer over TAP to mimic legacy functionality. Once the simple service layer is complete, moving modular components of TAP services to new technologies allows us to move all interfaces to a data holding with less migration effort.

3. Stack Components and Modularization

At the lowest level, MAST infrastructure runs on a combination of Linux and Windows Server platforms, both of which are supported by load balancing and failover architecture. Changes to the database, web server, and application layers in the near term can therefore be made with a reliance on either.

3.1. Database Platforms

The existing database layer for catalogs served by VO infrastructure is Microsoft's SQL Server. Built commercially to handle large data volume, MSSQL holds catalogs such as Pan-STARRS PS1 DR1 at 10 billion rows, using distributed database features hidden at the software connection level. MSSQL supports spatial queries with internal indexes (Katibah & Stojik 2011), used heavily in astronomical queries based on areas of the sky. MAST web applications call stored procedures within the database server using these indexes, keeping computation closer to the data for improved performance. MSSQL integrates with a Resource Governor tool limiting CPU and memory usage, critical to mitigating accidental client attacks. Given the value of these features for large and heavily-used catalogs, we will maintain this in the near term.

While MSSQL meets MAST's needs for existing catalogs, other database technologies have usefulness in a VO service stack. PostgreSQL with pgSphere for spatial queries has recently proven sufficient for large astronomical catalog services including Gaia, and is an open technology. Relevantly for TAP, the Astronomical Data Query Language (ADQL) syntax for spatial queries is much closer to that of pgSphere by design (Ortiz et al. 2008). The translation layer must handle this and NATURAL joins, natively unsupported in MSSQL; both features are a substantial effort in the service layer. A move to PostgreSQL allows translation layer simplification, potentially improving performance in some cases and removing opportunities for errors. PostgreSQL support also could allow closer to out-of-the-box TAP and simple service support for small community-contributed HLSPs.

3.2. VO Web Services

MAST supports several web service stacks with an eye to future consolidation. Legacy PHP simple services and some C# services can be moved to any new stack by connecting any TAP service on that stack to their data and having a simple service layer query

TAP and pipe back output. The existing TAP service on CAOM can be used in some cases, and we can add TAP access to more mission-specific data than defined in that model (Dowler et al. 2007).

Current MAST TAP infrastructure runs on IIS and has several modular components, some of which are collaboratively developed at or with IVOA partner institutions. Using the tool IKVM (Frijters 2014), components developed in Java can be run as libraries in other .NET software. This lets us run a Java-based open source ADQL translation module with MAST-contributed MSSQL-related code. The larger ‘taplib’ project (Mantele 2018) containing the ADQL translation module, which knows several SQL dialects, is in use for small TAP services in the IVOA, is being actively maintained, and it may be usable in full with some modification for MAST’s needs. Other open TAP implementations in Java also exist. The other major TAP module, currently in C#, is a Universal Worker Service (UWS) library managing asynchronous jobs, which accesses networked storage to provide job and result access across service instances. A general request handler module manages RESTful endpoints to run synchronous queries and pass async ones to UWS. A series of configuration files drive most automation; they include connection information and content for service information and example queries in HTML providing catalog-specific and data model use cases. Included in the configuration is SQL scripts for generating schema metadata information the TAP service uses to declare itself to clients and validate incoming queries.

Externalizing configuration settings increases potential automation and testing in each rollout of a catalog and its VO services. On startup, a TAP service builds an internal model of its schema from database tables and validates it. This schema information can be used by automated tests, as can the exposed example queries, which can in turn test the spatial indexes. UWS and network storage can be tested with an asynchronous call, all from a very simple client leveraging Astropy functionality, either in Astroquery or PyVO. Further automation will be achieved by tagging tables and columns used for spatial queries within in the TAP schema and modifying the software to detect them; this work will be based on UCDs as in some existing MAST simple services.

3.3. Clients

While clients are not a part of the web service technology stack itself, they are a necessary component of the web workflow, and another place where MAST can leverage IVOA-related projects in open development. Due to MAST’s roles in enabling research and pipeline and instrument analysis, we must work to support both functional and research-based clients. Current common clients to VO infrastructure include the MAST web portal itself for some simple services, Java-based GUI TOPCAT and its command-line STILTS, simple shell scripts for automated pipeline checking from the Canadian Astronomy Data Center (CADR), and the Astroquery utility TAP/TAP+, developed as a part of the Gaia mission with full IVOA interoperability in mind (Ginsburg et al. 2018; Segovia 2016).

We will expand MAST Portal support to include direct TAP access for research and functional use and continue testing with TOPCAT/STILTS, particularly for standards validation. The next phase of work for TAP architecture includes continuous integration testing in Python. MAST has developed and will continue to create Jupyter notebooks for TAP-based use cases involving our holdings, using the TAP/TAP+ utility; these can also be run in an integrated test environment. MAST is engaged in the

Astropy / Astroquery Python community and will continue to be more so as TAP and simple services are updated and used with these libraries.

4. Conclusions

Building a new, modular web service infrastructure for data discovery and access based around the Table Access Protocol IVOA standard helps further MAST's mission to develop and support user-friendly and scientifically useful search tools and foster inter-archive communication and interoperable standards. Building to this standard allows us to migrate legacy services for distributing our catalog, image, and spectral data, and update these as well as directly exposed TAP services with less future effort. Exposing MAST metadata directly through the CAOM data model and TAP standard API simplifies data integrity checking in inter-archive data mirroring pipeline, provides scripted metadata checking for instrument scientists, and promotes research through popular Python-based client tools. MAST already leverages open source components for TAP infrastructure and contributes to those projects and the standards themselves, fostering partnership and improving archival research software across institutions. Continuing to contribute to and integrate these components will help us keep our technological stack updated while meeting the unique technical challenges of the MAST archives and enabling multi-mission, cross-archival research.

References

- Dowler, P., et al. 2007, Common Archive Observation Model, Version 1.0, Tech. rep., National Research Council Canada
- 2011, IVOA Recommendation: Table Access Protocol Version 1.0, Tech. Rep. Version 1.00 (arXiv:1110.0497)
- Frijters, J. 2014, Ikvm.net. <https://www.ikvm.net>
- Ginsburg, A., Sipocz, B., Parikh, M., Woillez, J., Groener, A., Liedtke, S., Robitaille, T., Deil, C., Norman, H., Svoboda, B., Brasseur, C. E., Tollerud, E., Persson, M. V., Seguin-Charbonneau, L., Armstrong, C., de Val-Borro, M., Morris, B. M., Mirocha, J., Yadav, A., Seifert, M., Droettboom, M., Moolekamp, F., James Allen, Bostroem, A., Egeland, R., Singer, L., Rol, E., & Grollier, F. 2018, astropy/astroquery: v0.3.7 release. URL <https://doi.org/10.5281/zenodo.1160627>
- Katibah, E., & Stojik, M. 2011, New Spatial Features in SQL Server 2012, Tech. rep., Microsoft
- Mantele, G. 2018, taplib. <https://github.com/gmantele/taplib>
- Ortiz, I., et al. 2008, IVOA Standard. <http://www.ivoa.net/Documents/latest/ADQL.html>
- Segovia, J. C. 2016, Tap/tap+. <https://astroquery.readthedocs.io/en/latest/utils/tap.html>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Application of Google Cloud Platform in Astrophysics

Marco Landoni,¹ G. Taffoni,² A. Bignamini,² and R. Smareglia²

¹*Istituto Nazionale di Astrofisica, Osservatorio Astronomico di Brera, Via E. Bianchi 46, Merate (LC) - ITALY; marco.landoni@inaf.it*

²*Istituto Nazionale di Astrofisica, Osservatorio Astronomico di Trieste. Via G. B. Tiepolo, Trieste - ITALY*

Abstract. The availability of the new Cloud Platform offered by Google motivated us to propose nine Proof of Concepts (PoC) aiming to demonstrate and test the capabilities of the platform in the context of scientifically-driven tasks and requirements. In this poster, we review the status of our initiative by illustrating 3 out of 9 successfully closed PoC that we implemented on Google Cloud Platform. In particular, we illustrate a cloud architecture for deployment of scientific software as a microservice coupling Google Compute Engine with Docker and Pub/Sub to dispatch heavily parallel simulations. We detail also an experiment for HPC based simulation and workflow executions of data reduction pipelines (for the TNG-GIANO-B spectrograph) deployed on GCP. We compare and contrast our experience with on-site facilities comparing the advantages and disadvantages both in terms of total cost of ownership and reached performances.

1. Introduction

Google Cloud Platform offers a variety of services, ranging from storage to high performance computing and workflow execution, that could be exploited in the context of Computational Astrophysics. In this paper we review three Proof of Concept (PoC) out of the nine proposed to Google that have been successfully implemented on the public platform illustrating the architecture and the main results we have obtained. The paper is organized as follows: in Section 2 we illustrate the PoC for HTC oriented application while in Section 3 we comment on the implementation of HPC cluster on the Google Cloud Platform. We also tested the execution of Workflows aiming to offer instrument pipeline as a service reporting the results in Section 4.

2. PoC 1 - HTC Workload on the Cloud. The case of DIAMONDS

DIAMONDS (Corsaro & De Ridder 2014) is a Bayesian inference code that is designed to process data from astroseismology, a technique to study stellar oscillations through photometry or spectroscopy in order to derive their internal structure and physical parameters, such as the true mass. DIAMONDS has demonstrated (on premises) to be runnable in parallel through *embarrassingly parallelism* paradigm with almost no network communication. This kind of computational approach is very suitable

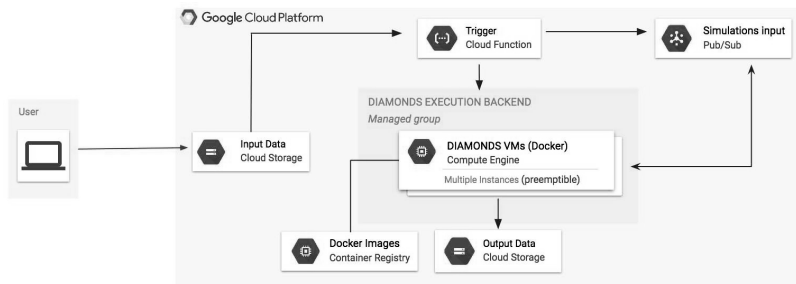


Figure 1. The HTC serverless architecture for DIAMONDS pipeline (PoC1)

to test HTC workloads on GCP. We introduce in this scenario a concept of *serverless HTC scheduler* that fruitfully exploits a heterogeneous set of GCP components such as Pub/Sub, Cloud Functions and Managed Groups in GCE (see Figure 1). Typical resource schedulers are complex middleware that have to deal with a limited amount of resources available accordingly to a set of time-sharing policies. An advantage of our approach is to use some equivalent concepts such as the queue from the available services to manage the execution of HTC workloads while guaranteeing a general purpose approach to many possible HTC computation. In the architecture that we design, keeping in mind to be as much as general as possible, the computation starts by uploading to Cloud Storage a plain text file that contains, for each row, the data necessary to perform a single simulation. These rows are pushed, by a triggered Cloud Function, into a Pub/Sub topic. Then, a cluster of instances (regular or preemptible) dimensioned runtime is fired up accordingly to the estimated size of the whole workload. Each node of the cluster, after starting up with a pre-configured Image on Compute Engine, pulls a number of messages (proportional to the number of vCPUs available) from the PubSub queue starting the computation of various DIAMONDS simulations using Docker containers. Data produced locally by DIAMONDS on each instance are finally transferred to a bucket on Google Cloud Storage before shutting down. This method allows to deploy an HTC-based architecture, suitable for many projects that share the same kind of parallelism and requirements on the workload, that scales both vertically (number of cores per node and thus number of simulations) and horizontally through an elastic cluster fired up accordingly to the number of required simulations and CPU/hours.

3. PoC 2 - Exploring HPC capabilities with Google Cloud Platform

GADGET (Springel 2005) is a lagrangian code to perform numerical simulations of gravitationally interacting particles of both dark matter and baryonic matter which computes gravitational forces using a TreePM technique. A mean field approximation is used for large scales (Particle-Mesh, PM) while at smaller scales a usual Treecode is used. Hydrodynamics is solved using a so-called Smoothed Particle Hydrodynamics technique. GADGET is an HPC code based on message passing interface (MPI) libraries and OpenMP. It is written in C and requires some support libraries to run (FFW2.4, HDF5, GSL). To test the performance of the Google Cloud infrastructure for HPC applications, we executed a virtual cluster managed by Slurm scheduler composed

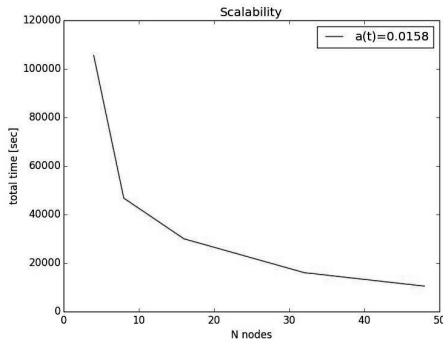


Figure 2. GADGET scalability on GCloud.

of both reserved nodes and On-demand (non-preemptible) instances. To automate the deployment of the cluster, we use the Cloud Deployment Manager (CDM). This service aims to automate the creation of complex resources and services where various entities are described in terms of *yaml* files and deployed using *gcloud* command line interface. In this PoC, we modified the Slurm official CDM files to modify the resources (we used 4 cores with 4 GB ram per core) and the software (we added MPI, FFTW, HDF5, and GSL). Moreover, we deployed a cluster where only two computing nodes are running and more resources are bootstrapped on demand using Slurm only if necessary.

We tested GADGET scalability with a small cosmological BOX of 778,688 particles for a Λ CDM model ($\Omega_0 = 0.24$, $\Omega_\Lambda = 0.76$, $h = 0.72$) increasing the number of nodes and the size of nodes (up to 96 cores and 624GB Ram). We present our scalability results in Figure 2. The GCP infrastructure is based on standard ethernet connections, while for HPC applications the role of a low latency high throughput interconnect is crucial, as evident from Figure 2. On the other side, the cluster is suitable to any HTC applications where the inter-node communication is not present or limited.

4. PoC 3 - Workflow execution. Running GIANO-B data reduction pipeline as a service

In this use case we report and comment about the creation of a scaled and balanced environment, whose purpose is the execution of workflows submitted by the user through the workflow environment *Yabi* (Hunter et al. 2012). This scenario involves the user, who retrieves GIANO-B raw data from TNG archive public and private storage, and the execution of the GOFIO data reduction pipeline to produce reduced data that can be retrieved by the user. The main aims of this PoC involves the simplification of the management of the infrastructure, moving from an on-premises infrastructure to PaaS/SaaS layers offered by Google Cloud and of the deployment of software using containers to avoid incompatibility issues between packages that must coexist and work together. Finally, this PoC aims to improve software and service maintenance while optimizing and balancing the scalability of the service according to the load. The implementation of this PoC foresees these services from GCP: Google Compute Engine for virtual machine instances management, Slurm or Google Kubernetes Engine as workload

manager to deploy GOFIO container and the Docker platform for the containerization of GOFIO pipeline. Since we have two workload managers, two different solutions for this PoC was implemented. In the first architecture we made use of *Yabi* and Slurm while in the second one we exploit *Yabi* coupled with Kubernetes. For both architecture *Yabi* was deployed on a Compute Engine instance that acts as frontend for final user. Slurm cluster was deployed using standard *yaml* file available through Slurm official documentation¹. To connect *Yabi* with Slurm, we used the native *Yabi*-Slurm backend connector, which is available in the latest version of *Yabi* (version 9). Kubernetes cluster was deployed using Kubernetes Engine following the official Google documentation² deploying a Network File System (NFS) server from Cloud Launcher, configuring Persistent Volumes, POD ReplicaSet, LoadBalancer and HorizontalAutoScaler. *Yabi* does not provide a default backend connector for Kubernetes, therefore we used the default *Yabi*-SSH backend connector to connect *Yabi* to Kubernetes cluster generating SSH key in *Yabi* instance and adding it in the Kubernetes cluster. To test the performance of both architecture and to check actual scalability as function of the load, massive tests submitting simultaneously tens of jobs were performed. As a reference, for on-premises infrastructure these large workloads result in an excessive dilation of the execution times, since the total execution time of all the jobs (submitted simultaneously) is much greater than the sum of the execution times of the individual jobs performed one by one and, in most extreme cases, *Yabi* crashes. For the architecture *Yabi*-Slurm deployed on GCP the scalability is good and all jobs are completed correctly with no significant time leaks compared with the execution time of a single job. Slurm is natively supported by *Yabi* and it performs reasonably well in managing the job queue and the scaling. New Compute Engine instances are created and destroyed on demand efficiently according to the load. However, for what concerns the *Yabi*-Kubernetes the scalability is also remarkable, but some jobs (about 1 each 8) exit with error and they are not more recovered, probably due to the fact that in this configuration the job queue is completely managed by the *Yabi* SSH Backend that submit jobs to Kubernetes which seems able to manage the load, but the *Yabi* Backend fails to manage all the job queue. We evaluate a total estimated charges of about 200 EUR/month to maintain both architectures up and running.

References

- Corsaro, E., & De Ridder, J. 2014, AA, 571, A71. 1408.2515
Hunter, A., B Macgregor, A., Szabo, T., A Wellington, C., & I Bellgard, M. 2012, Source code for biology and medicine, 7, 1
Springel, V. 2005, MNRAS, 364, 1105. astro-ph/0505010

¹<https://github.com/SchedMD/slurm-gcp>

²<https://cloud.google.com/kubernetes-engine/docs/how-to/creating-a-cluster>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Transforming Science Code into Maintainable Software, Insights into the G-CLEF Exposure Time Calculator (ETC)

Charles Paxson, Joseph B. Miller, and Janet D. Evans

Harvard-Smithsonian Center for Astrophysics, Cambridge, MA, USA;
cpaxson@cfa.harvard.edu

Abstract. We explore a common workflow in research institutions where science code is transformed into robust, maintainable, and expandable code. The case study presented is the Exposure Time Calculator (ETC) for the Giant Magellan Telescope Consortium Large Earth Finder (G-CLEF) spectrometer. We describe the process we took to develop functional requirements documentation and a web application from science code. The ETC provides a rich set of features to help the scientists estimate the performance of the instrument including: the computation of exposure time, SNR, and precision radial velocity, GUI text results, downloadable FITS standard compliant summary of results, and graphical displays. We highlight the importance of a functional requirements document for information exchange between scientist and engineer, where principles and assumptions can be collaboratively understood and solidified. As the document matures, scientists may use it as the basis to specify new requirements. We discuss the importance of making physical interpretations of the code, of understanding and ultimately cleaning science code magic numbers, and of comprehending the overall flow. This detailed analysis is important since requirements morph as the project progresses.

1. Introduction

In this paper we discuss the development, design and use case of the G-CLEF Exposure Time Calculator (ETC) from extracting the science specification and software requirements from a bare bones science prototype to the implementation and deployment of a web application. An ETC is a commonly available analysis tool for a telescope and instrument system that supplies feasibility feedback to an astronomer's scientific observation goal. We acknowledge that prototype software written by scientists is a common practice to communicate requirements. However, maintainable software needs a functional requirements document, so we highlight the importance and discuss the process of creating a functional requirements document for information exchange between scientist and engineer, where principles and assumptions can be collaboratively understood and solidified. We present the ETC design and describe a number of the features of the ETC, especially those that were developed beyond the science code delivery and layout the process of scientific validation.

The G-CLEF ETC has been developed to support the Giant Magellan Telescope (GMT) cross-dispersed echelle spectrograph operating in the optical to NIR wavelength bands. G-CLEF is the GMT first light instrument with the primary mission to detect and characterize low-mass exoplanets in orbit around solar-type stars by the method of precision radial velocity (PRV) measurements. In addition, the spectrograph is designed

with sensitivity suitable for stellar abundance studies and for observing astronomical objects at high redshift.

The G-CLEF ETC provides a rich set of features to help scientists estimate the performance of the instrument. Various models are allowed such as: stellar model, Power Law (i.e. Galaxy, AGN), or a user defined model. Source brightness is affected by the input parameters: source magnitude, and an embedded ISM extinction model. The redshift or radial velocity of the source is input. Instrument and Signal (sky brightness) noise is taken into account. The ETC monitors the CCD instrument effects such as pixel read capacities. The output includes: the computation of exposure time, the signal-to-noise (SNR) over all wavelengths, and in precision radial velocity (PRV) mode, the PRV is output as well as the total counts as a function of wavelength.

2. Importance of a Functional Requirements Document

As the software developer studies and understands the prototyped science code, it becomes imperative for them to develop a functional requirements documentation. There is often a drive to make progress and there is a tendency to further enhancements without recording the methods and assumptions. It is even possible during this process that new features are identified and requested to be added. However, it behooves all parties to take the time to document. From this anchoring point, where the physics and engineering principles and assumptions are revealed, there is enough shared knowledge to begin a clean design and bring the code to the next level of accuracy. The documentation will expose details that are not well understood, and the document becomes a useful means of communication between scientist and software engineer. The document serves the scientist so that they no longer need to dig through unfamiliar code to receive the understanding that they need to ensure the validity of the application. It is used by the software developer as a basis for functional requirements where it becomes an important component of a maintainable software application.

The baseline prototype code was a Python script that provided the component needed to transform the model data. With a model-view-controller (MVC) design that we adopted, it allowed separation of the science code into a more maintainable piece of software. The first steps in understanding the prototyped code was to map out the code in the form of a flow chart. This provided a clearer understanding of the what the code was doing and where the inefficiencies lie. For example, within the ETC, high fidelity stellar atmosphere models are convolved to the lower instrument grid resolution. The wavelength degradation process is relatively expensive to generate these spectra, and the flow chart showed that the spectra were being computed twice. Because of our documentation process and reconsideration of the basic equations underlying the ETC, we converted the spectra to a time dependent one and applied the exposure time as final step to eliminate the duplication.

These initial steps paved the way for a formal write-up of the science specification and software requirements that was extracted from the prototype code. The detailed functional documentation allowed additional refinements that were deemed necessary to develop more readily as it provided a more complete understanding of what we wanted to achieve from the code. We upgraded the ETC to include refinements by using the proper spectral and spatial distributions of energy on a resolution element, and we upgraded the ETC to monitor the read process - flagging cases where the pixel well capacity was exceeded.

2.1. Challenges of Science Prototype Delivery

Science code delivery as a form of functional requirements is a common practice to launch a project. Other possibilities exist such as documentation: descriptions of the algorithm, class diagrams, references, theory, and or flow diagrams. Delving into science code without the background or context is time-consuming and costly work. This challenge was compounded because in this particular case we did not have access to the science code author. Areas that can make the process longer include the fact that science code need not be robust to every corner case, or that bugs in the science code obfuscate the comprehensibility of the code. Inefficiencies such as the duplication of spectra generation are of little consequence to a science application, but such inefficiencies or slowness in response may undermine a positive user experience. However, understanding whether the inefficiencies are important can be difficult to determine. Derived quantities can become opaque in how they connect to the underlying physics and engineering principles and magic numbers create uncertainty in the meaning of code lines. Lastly without test cases, supporting descriptions or relied upon comments, the units of the quantities may be hard to decipher such as our case in determining whether our output was per pixel or per resolution element with confusing variable names that confounded the understanding.

2.2. Further Benefits in Driving Toward a Functional Requirements Document

Finding the discipline and patience to document is a challenging task. However, having a well defined science specification and set of software requirements have many very desirable benefits. Scientists with differing background and differing conceptual language can see the principles explicitly in equation form. New personnel assisting the project can quickly come up to speed. The document becomes versioned providing the history of the project. The document became the basis for a help page. The project can be safely transferred to our contractor if the need arises where the code can be easily managed and maintained.

3. G-CLEF ETC Web Application

For the G-CLEF ETC, we immediately adopted a Model-View-Controller (MVC) design pattern. The View component is comprised of html and javascript. We make use of the DiGraph library for user interactive scientific graphs that display the total counts and signal to noise as a function of wavelength results. The Model component is composed of objects that contain data, units, descriptions and FITS header keys that are extracted from JSON files. The Controller component is implemented with Flask, a backend Python microframework used for web development. The integrated science code, the component of the controller that transforms the model data, is encapsulated within the GCLEF_ETC class.

3.1. Use Case

The ETC provides a user the ability to assess the efficiencies of the instrument under different conditions. As such, the ETC was implemented early on in the process to help the engineering team analyze the capabilities of the instrument. It will later be used by the end users to plan their observation proposals.

3.2. Modular Design

The MVC modularized the overall web application. Further modularization was formalized within the transformed science code such as including a JSON input system to populate the data model. The calibration system is composed of engineering data tables such as the optical path efficiencies, the CCD detector quantum efficiencies, the band filters, and the resolution element energy distributions which are engineering parameters that are version controlled and isolated from the code.

3.3. Validation & Verification

The transformation of the prototype to maintainable and robust software requires validation and verification. Step by step unit tests are created and installed as part of the code base. For the G-CLEF ETC, the model spectra is a fundamental quantity from which exposure times and signal-to-noise are derived, and for this reason we compared our internally generated spectra against a set of standard stellar fluxes. Scientists familiar with other telescope systems of similar class devise comparison tests. Hand calculations are made, and analysis goes into certifying proper behavior at extremes which in our case include intensely bright and faintly dim stars.

4. Conclusion

From delivered science code we have developed an ETC web application that is available to the astronomy community for assisting observations with the G-CLEF spectrograph. The ETC is version controlled, calibration versioned, validated, and secured with unit tests. We advocate for the delivery of design specifications, but given that science code delivery as requirements is a common practice, we highly recommend documentation that assists design and enhancements during the development phase. The document serves as a record to assist future enhancements, inform new personnel that join the project, and provide a clear description of the software intent aiding smooth transfer to another agency for continued maintenance if the need arises.



Janet Evans, Ray Plante and Theresa Dower (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Centralization and Management of Science Operations Procedures and Test Cases using SOCCI

F. Pérez-López,¹ V. Navarro,² K. Lumi,³ H. Liiva,³ H. Hanson,³ R. Caballero,⁴ C. L. Kuik,³ A. Lember,³ C. García,⁴ E. Pašenkov, and ³L. Kaldamäe³

¹*RHEA for ESA, Villanueva de la Cañada, Madrid, Spain*

²*European Space Agency (ESA), Villanueva de la Cañada, Madrid, Spain*

³*CGI, Helsinki, Finland*

⁴*Econocom, Madrid, Spain*

Abstract. The Science Operations Configuration Control Infrastructure (SOCCI) is a single, highly customizable platform for system engineering providing tools and guidelines for: requirement management, problem and change management, test management, project and document management, source version control and continuous integration. This infrastructure provides support to the software development and maintenance processes of science operations units at the European Space Astronomy Centre (ESAC). SOCCI reduces effort and knowledge to setup & maintain the Systems Engineering Environment (SEE) and supports the users by providing guidelines and good practices learned from previous experiences. The development of SOCCI started in 2014 and has been operationally used from June 2017. Recently, the range of functionalities already covered by SOCCI have been extended through SOCCI Evolution and SOCCI Test Framework projects. This paper describes the design, implementation and use of SOCCI for two major new functionalities: the management of operational procedures and documentation including their scheduling and automatic execution; and the testing automation including the importing and exporting of results from external test tools.

1. Introduction

Three different organizational units at ESAC are involved in configuration control activities for science systems in Science Operations Department (SCI-O). Over the years, these units have utilized several systems to support development and operations of science systems throughout their lifecycles. Evolution of working practices and different project needs have resulted in the coexistence of multiple tools in order to fulfill configuration management requirements. Based on the analysis of the existing working practices and systems in use at ESAC, a single framework covering all the requirements was developed.

The Science Operations and Configuration Control Infrastructure (SOCCI) is a single, highly customizable platform for system engineering, providing tools and guidelines for requirement, problem and change, test, project and document management, source version control and continuous integration. This infrastructure provides support the software development and maintenance processes of science operations units at ESAC

adopting the guidelines outlined in ECSS-E-ST-40C ¹ standard for software development as the reference process framework to validate adequacy of the configuration management framework to current practices.

SOCCE reduces effort and knowledge to setup & maintain the Systems Engineering Environment (SEE) and supports the users by providing guidelines and good practices learned from previous experiences. The use of common tools facilitates the knowledge transfer across projects and smooth the learning curve when personnel are moving from one project to another. SOCCE simplifies the configuration management procedures and reduce the list of applications previously used at ESAC, and requires less effort and knowledge for maintenance and supporting the users.

1.1. SOCCE Lifetime

The development of the SOCCE started in 2014-2015 as part of the ESA Geo-Return initiative. The project was developed by CGI Finland and from June 2017 is fully operational being now under maintenance phase. Since becoming operational, its use has become quite large and more projects are moving to SOCCE (currently 52 projects, 390 users and 30000 issues). As more people get involved with the system, new features and changes are requested. Furthermore, the needs and requirements of the system evolve and change over time.

From July 2018 two new major projects to enhance SOCCE have been started with this purpose: SOCCE Evolution and SOCCE Test Framework which basically extends SOCCE in order to be able to cover science operations needs for project management, operations management, and testing management and automation. The objective of this paper is precisely to describe the two latest functionalities.

1.2. SOCCE Main Features

SOCCE is fully modular and is organized around the following areas (ordered by maturity implementation level from high to low): requirements management, problem and change management, project management, document management, source code management, release management, test management, system design, system help desk and lately operations management. SOCCE's tooling trade-off assessment has been driven by SCI-O and industry best practices in order to define an evolutionary adoption path. SOCCE makes use of different tools for each of the engineering areas: Atlassian JIRA , Atlassian Confluence, Atlassian Bitbucket, Nexus Sonatype, Jenkins, Zabbix, Splunk, Sonar, Kayako, etc. SOCCE can be used in two different ways, depending on the user needs:

- **as a Service:** The project is hosted in the SCI-O infrastructure at ESA/ESAC and JIRA/Confluence/Bitbucket/Nexus applications are maintained by SCI-O team. Request for changes/Resolution of bugs are managed through SCI-O SOCCE CCB process)
- **as a Package:** The project is hosted by the project and then, SOCCE applications are maintained by the project. SOCCE plugins are provided to the project and a fork of the source code is also provided. This configuration is available for ESA internal projects and requires a license agreement for non-ESA projects. The

¹<http://ecss.nl>

project is free to install and make changes to the SOCCI plugins based on their needs, keeping full control and governance on their SEE.

2. Testing management

SOCCI is being highly improved in the area of testing management. The current version of SOCCI only includes the definition and classification of the tests and allows the automatic generation of test specifications and test reports. The new version of SOCCI will include some improvements:

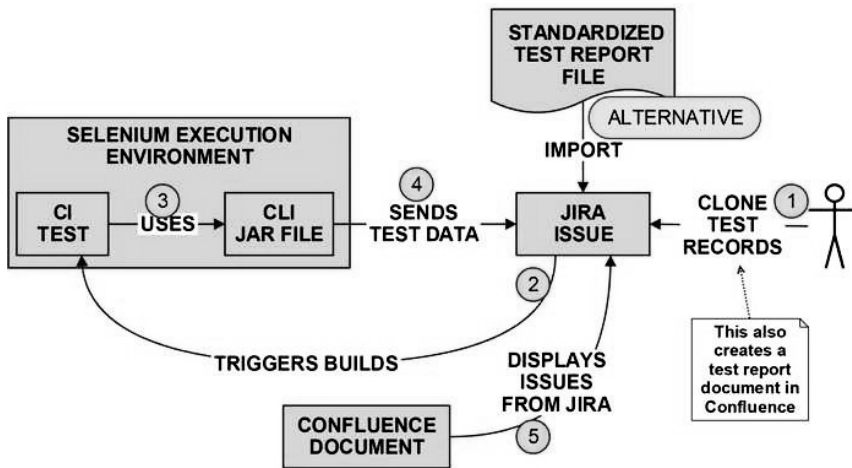


Figure 1. SOCCI Testing Management Approach

- **Testing Command line interface:** This tool will allow to read and update test records from command line. It is based on a API which could be integrated with specific project testing tools (e. g. Cucumber, Selenium, Robot framework) providing the possibility to report the results of the testing executions into JIRA test records including the test execution status.
- **Test Reports importing:** This will allow to import, automatically or manually, the output of specific project testing tools to JIRA as test records. This functionality isolates the testing documentation from the used technology by facilitating the communication between testing teams across SCI-O units.
- **Automation of test Execution:** This function will allow to launch from JIRA the execution of automatic test procedures. It is possible by combining the test report importing and the use of continuous integration tools (e.g. Jenkins).

3. Operational procedures management

SOCCI will include a new module based on JIRA/Confluence to allow the definition, scheduling and reporting of specific operational procedures. Three issue types have

been defined to cover all the possible use cases: ‘procedure specifications’ which groups the operational procedures by different criteria, ‘procedure definitions’ which includes the procedure steps, inputs, constraints and outputs and the ‘procedure records’ which are the instantiation of procedure definitions at each operational campaign. The reporting and documentation of the procedure specifications and execution reports will be done using Confluence, in a similar way it is used for test procedures. The visualization of procedure record status for each operational campaign will use the JIRA Kanban boards and the automation of the execution will use the same mechanisms used in SOCCI for testing automation. In addition to the previous, Operations Change Requests (OCRs) have been defined to track problems and improvements during operations. They could be configured to allow multiple authorization, which is a relevant functionality with operations between different control centers.

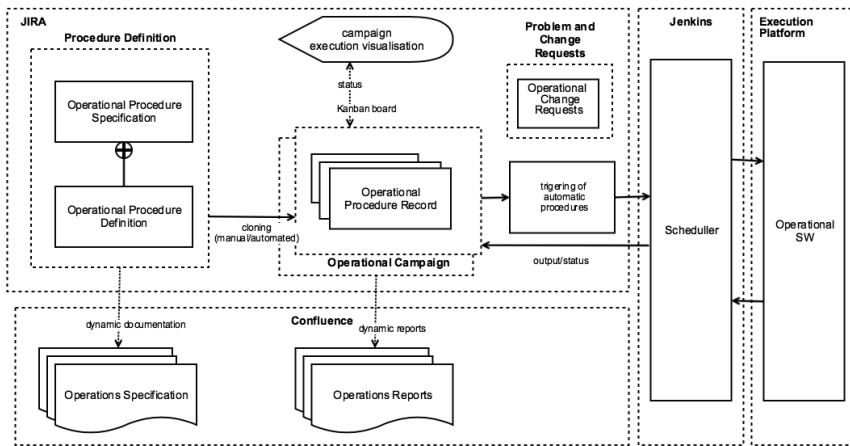


Figure 2. SOCCI Procedure Management Approach

4. Conclusions

SOCCI is a consolidated infrastructure for ground systems engineering and provides support across ESAC organizational units. Next steps for SOCCI have been described in this paper and cover two major gaps in all configuration management systems: the test specifications management and automation and the operational procedures management. These new functionalities consolidate SOCCI as reference engineering tool for ground segment development and operations.

Session VIII

Databases and Archives: Challenges and Solutions in the Big Data Era

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Astronomical Archives: Serving Up the Universe

Felix Stoehr

ESO/ALMA, Garching bei München, Germany; fstoehr_eso_org

Abstract. In this review, we first briefly look at some of the current context of storing and making astronomical data discoverable and available. We then discuss the challenges ahead and look at the future data-landscape when the next generation of large telescopes will be online, at the next frontier in science archives where also the content of the observations will be described, at the role machine-learning can play as well as at some general aspects of the user-experience for astronomers.

1. Science Archives

1.1. Rationale

The main purpose of Science Archives is to enable astronomers around the world to undertake scientific programs by making use of observations that have already been carried out, especially as multi-wavelength astronomy is more and more popular. With those data being available for free, they also serve as a great resource for astronomers in developing countries who may not have access to telescope time. Astronomical Science Archives are however also intensively used in the proposal process for duplication checking, for studies of time-variability but also for citizen-size and in outreach.

Last not least, it is the foundation of science to be able to reproduce the results and Science Archives provide the long-term persistence of the corresponding data - i.e. measurements of photons.

1.2. Photons

Astronomy is largely concerned with photons which are astonishingly simple particles. Photons can originate from a certain *position*, they can possess a certain *energy*, they can arrive at a certain *time* and they can be *polarized*. While light-beams in principle also can carry orbital angular momentum, no successful measurements have yet been carried out. The complexity of measurements themselves is much smaller than the complexity in other disciplines of science like for example Biology. This is also true for the measurements of the other astronomical messengers, Neutrinos, Cosmic Rays and Gravitational waves which essentially have the same properties as the photons.

While the raw data-products of astronomical observations can be relatively complex and in-homogeneous - interferometric visibilities, wrapped Echelle-spectra, γ -rays induced distributions of secondary photons - the reduced data-products are simple, reflecting the simpleness of the photons. Essentially all science-grade data can be stored in six-dimensional arrays (Fig1), with two dimensions for the position (e.g. RA/Dec), one for the energy (e.g. frequency), one for the time (e.g. UTC), one for the polariza-

Pos1	Pos2	Energy	Time	Pol.	Quantity	
1	1	N	1	1	1	Spectrum
1	1	1	N	1	1	Time-series
N	N	1	1	1	N	Image with error map
N	N	N	1	N	1	Data cube with polarisations

Figure 1. Photons are simple. The science-grade data of essentially all astronomical measurements can be stored in six-dimensional arrays where some of the dimensions might be degenerate depending on the type of data like spectra or images. The Quantity contained in the 6D hyper-cube can be e.g. photon counts or flux or errors or weights.

tion measured (e.g. Stokes I) as well as one for the type of measurement contained in the array (e.g. flux). Some of the dimensions might not be needed for a particular data and thus the dimension in the 6D hyper-cube is degenerate.

1.3. Photons in the Universe

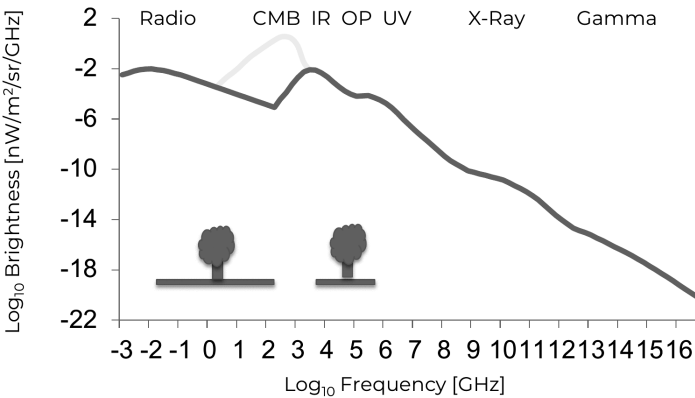


Figure 2. The brightness of the extra-galactic background light (EBL) as a function of the energy of the photons. The shown quantity is proportional to the number of photons in the Universe at a given frequency.

Fig. 2 shows the brightness of the extra-galactic background light (EBL) as a function of the frequency of the photons in the Universe (values from Hervé Dole¹). The brightness values are proportional to the number of photons. The largest number of photons originate from the last-scattering surface about 300,000 years after the Big

¹https://www.ias.u-psud.fr/dole/astrophysics/Dole_EBL_20130919_CASPAR_DESY_EBL_hdole.ppt.pdf

Bang and form the Cosmic Microwave Background (light-grey line). The other photons are subsequently emitted by astrophysical processes related to galaxy- and star-formation (dark line). γ -ray photons are rarer by at least 12 orders of magnitude than radio-photons. Not all photons reach the ground of the Earth for easy observation. The atmosphere is only transparent to photons in two windows, the optical and the radio window. The combination of those factors has the implication that the data-rates of the reduced science-grade data of astronomical experiments is largest in the low-frequency radio-regime. Indeed, with technology of storage and data-processing advancing still at exponential rates (but see section 2.1), challenges for the storage of science-grade data are currently and likely in the future only present for telescopes like LOFAR and SKA. The EBL of course only shows the average photon distribution in the Universe as a whole. Observations of bright objects, with the Sun as an extreme case, provide large amounts of photons following different spectral energy distributions.

1.4. Best practices for Science Archives

Over the last decades, astronomical Science Archives have been slowly but constantly improving and a number of Best Practices have emerged: Science Archives shall allow users to query the holdings by *physical parameters* rather than for example only observation-ID. They also shall provide full access to the entire parameter space without requiring a particular constraint (e.g. the position) to be placed (*unscoped searches*) (see also Stoehr (2017)).

Ideally, Science Archives also allow combined searches on metadata from the observations but also from the proposals as well as on metadata from the publications made with the facility's data. Users often need to query for an entire set of sources and thus providing a *target-list upload* has become standard. Today users have acquired a lot of web-habits which can be leveraged with interfaces providing a *modern user-experience*. At the same time, mainly driven by the increased data-rates, *programmatic access* to astronomical metadata and data, in particular through Virtual Observatory (VO) protocols, are mandatory. For highly-performant interfaces and in the light of transparency, it turns out that policies allowing all metadata to be shown on the interfaces to be *public* are preferable.

Next to the long-established *result tables*, modern user-interfaces feature small *previews* of the data-products as well as the footprint of the observation on a *sky view*. The AladinLite software package (Bonnarel et al. (2000), Boch & Fernique (2014)) and HiPS background display (Fernique et al. 2015) are here the current gold-standard. All data that are not protected by a proprietary-period any more should be downloadable without users having to create an account (*anonymous downloads*). If the data-sets are large, *parallel downloads* should be offered to make use of the entire network-bandwidth the users have available.

The data-products themselves should be fully calibrated and reduced and thus be *science-grade*. Also, the products should be *self-describing* (e.g. FITS headers) given that users might retrieve products with VO tools and never passing by the observatory's own interfaces where additional documentation might be available. The raw-data as well as the *Pipeline software* to allow users to recreate the science products need to be available. Data-products stored in the Science Archive should be reprocessed as soon as the Pipeline software has significantly changed.

Finally, it turns out that it is very useful to set up policies that require authors of publications making use of the observatory's data to cite that use including the corre-

sponding data-identifier (Stoehr et al. 2015a). This allows the observatories to close the loop between data and publications and provides the metadata needed to offer simultaneous searches on observations, proposals and publications.

1.5. Science Archive usage

An analysis of the query behavior of the Science Archives, here the queries on the ALMA Science Archive (Stoehr et al. 2017) for Proposal Cycle 6, is shown in Fig. 3. The results follow the findings of an analysis previously carried out CADC (Stephen Gwyn, private communication). Counting the number of occurrence of a particular query field in the queries to the ALMA Science Archive, we find that 80% of the total number are covered by only six query fields, 90% are covered by nine query fields. About 1% of the queries did not specify any query constraint and users did get the entire result set returned, for example to use the sub-filtering capability.

As with the exception of 'redshift', 'line transition' and 'object type' query fields, essentially no additional query fields have been requested more than by one user in ALMA's User Surveys, the present set of query fields is nearly feature-complete. The most important query fields are also those covered by the ObsCore data-model of the IVOA (colored background).

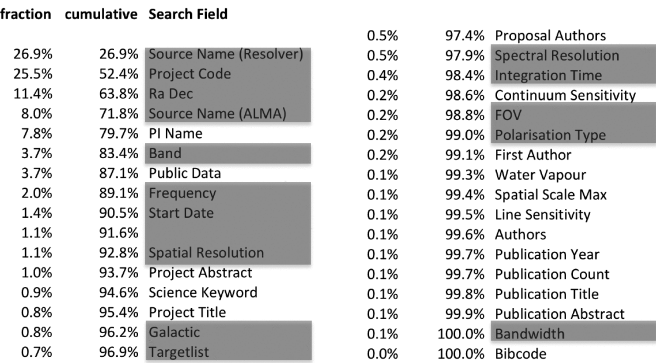


Figure 3. Usage of query fields in the searches to the ALMA Science Archive. 80% of the total number of query fields used are covered by only six query fields, 90% are covered by nine query fields. The colored fields indicate concepts covered by the IVOA ObsCore data-model.

2. Observatories

The evolution of Science Archives is of course very closely linked to the evolution of the Observatories themselves. For Observatories, we have been observing a number of trends in the last decades. We see a general move from closed, institute-driven experiments to open often internationally run observatories (e.g. ALMA, CTA and SKA). We also observe a concentration of Science Archives at data-centers who provide data-portals encompassing all the holdings, often in addition to the facility-specific interfaces (e.g. CADC, MAST, ESA and ESO). The development of data-portals is also helped by the wider adoption of metadata standards and protocols, in particular those

of the VO. An other trend is that observatories by now seem to have generally accepted the approach to deliver science-grade data to the users. Finally, as a consequence of the enormous advances in technology and astronomy, observatories are used to handling massive datasets.

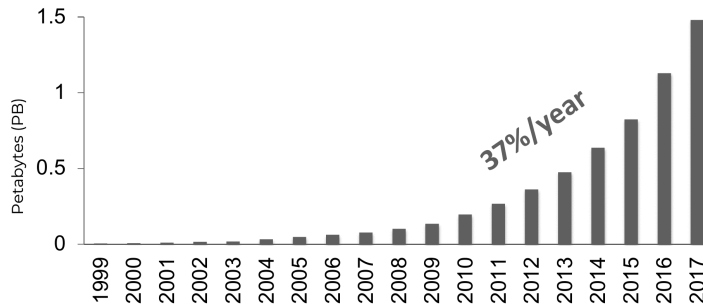


Figure 4. Petabytes of scientific data stored at ESO as a function of time (values courtesy of Adam Dobrzynski). The amount is rising exponentially at a quite constant rate of 37%/year.

As an example, Fig. 4 shows the amount of scientific data in PB stored at ESO as a function of time. We observe a surprisingly stable exponential growth of the data-holdings of 37%/year over nearly two decades in time.

2.1. Hard-disk cost

With hard-disk prices having dropped exponentially by about 40% per year - the so-called "Kryder's Law"² - storing the data was not overly challenging. Collecting available data of hard-disk costs³, however shows clearly that Kryder's law is broken since about 2010. The hard-disk prices since then have been dropping at a much slower but still constant rate of about 15% per year. As a consequence, hard-disk prices are more than a magnitude larger today than predicted by Kryder's law.

Unless the data-intake rate is changed, observatories will thus have to spend about 20% more on hard-disk storage each year. While this increase is not challenging for a very large fraction of observatories, for observatories with very high data-rates, this evolution should be closely monitored.

To illustrate the situation, Fig.5 shows the cost and physical-size evolution of a Science Archive if a constant amount of new data is acquired each year. This is the case for example for survey telescopes or for space missions. Assuming a 6-year hardware replace cycle and the above-mentioned 15% price-drop per year, the total amount of storage cost for a running time of the facility of 30 years is about nine times the storage cost required to store the first-year's data. Similarly, the physical size of the Science Archive servers will grow to over four times the size required to store the first year's

²https://en.wikipedia.org/wiki/Mark_Kryder

³<http://www.mkomo.com/cost-per-gigabyte-update> <https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte>

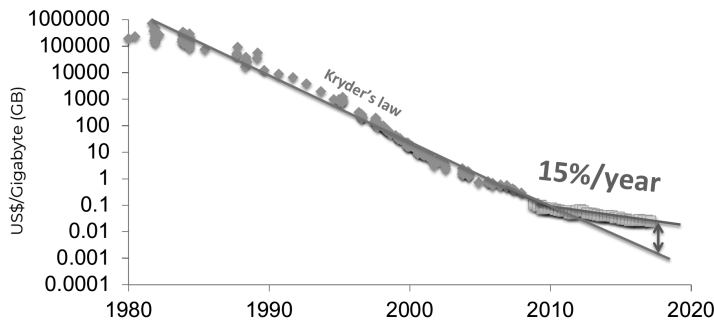


Figure 5. Evolution of the hard-disk prices with time in USD per GB. Kryder's law - a 40%/year price-drop is broken since 2010 and hard-disk prices only drop at a rate of about 15%/year.

data. These numbers are very significantly larger than the factors of 2.6 and 2.4 for the cost and physical size, respectively, when the Kryder's law was still holding.

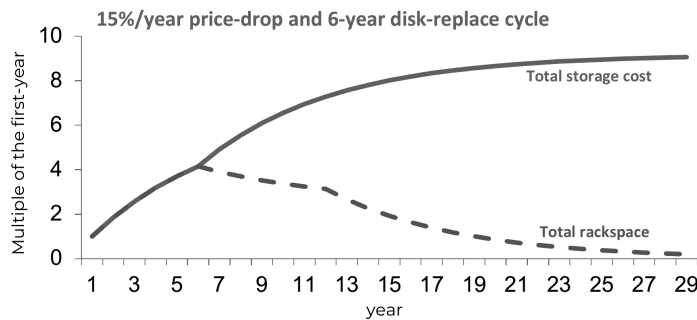


Figure 6. Evolution of the storage cost and the total rack-space required for a facility with linear data-intake (e.g. surveys or space missions) and a six-year hardware replace cycle. The total storage cost for a running time of the facility is nine times the storage cost required to store the first-year's data and the Archive will grow to over four times the size of the first year.

2.2. So much data

As discussed in the previous section, the amount of astronomical data available for research is growing exponentially. Indeed, today the VLT+ALMA+Magic produce about 70 GB/year/astronomer and the MWA produces already 350GB/year/astronomer of science-grade data products. If telescopes will create data as planned, then in the year 2030 the VLT+ELT+ALMA+CTA might produce about 1TB/year/astronomer and the SKA alone will provide 200TB/year/astronomer. For these back-of-the-envelope calculations the current number of astronomers registered with the IAU as well as a rough astronomer-doubling time of 10-15 years have been assumed. While the exact numbers are of lower importance, the message they send isn't: there will be far too much astronomical science-grade data available than humans who can look at them.

And the number of astronomer's does not scale, at least not fast enough (see also Stoeher et al. (2015b)). The fact that the amount of data can be handled technologically but not analyzed by humans any more is part of the "Big Data" paradigm.

Three solutions to the too-much-data problem are available: Alexander Szalay suggested to "Think of how to collect less data"⁴. This very elegant solution, focusing on less, but higher quality data, however, does currently not seem to be the trend in astronomy. The classical way forward is to process the data to ever higher levels. In some sense, this is what astronomy has been doing also in the past e.g. when the transition was made from delivering raw-data to users to delivering calibrated science-grade data products. Last not least, the advances in technology around Machine Learning, in particular Deep Learning, open the avenue that - at least for some problems - machines directly can help to produce scientific results from large amounts of science-grade data products.

2.3. Observatory responsibility

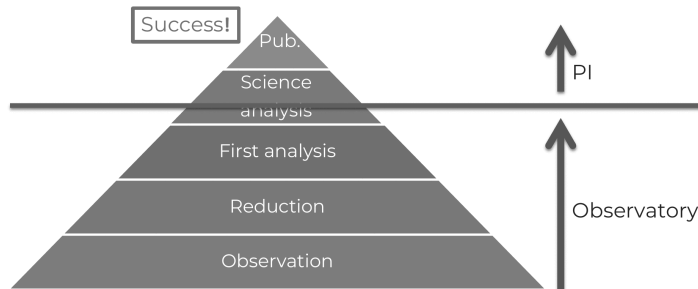


Figure 7. Expected future share of the workload from the observation to the scientific publication.

Whereas today's standard for PI-driven observatories is the delivery of science-grade data products, observatories will have to take a larger part of the process from the observation up to the publication (Fig. 7) and also provide higher-level products that cover the first- and even parts of the science-analysis. In short, they will need to provide for PI-observations data that are as useful as what survey telescopes (SLOAN, PanSTARRS, GAIA, LSST, etc.) have provided or will provide for their users. This will increase the responsibility of observatories in the scientific process and will come at a non-negligible cost (see Fig. 8). However, for observatories this step will be crucial to assure that PIs and archival researchers work with the data and convert them into science, which is the measure of an observatory's success.

One additional challenge is that more and more observational astronomical data are not spectra or images but more than three of the dimensions of the 6D-hyper-cubes of section 1.2 are filled. In particular many of the future instruments at the VLT (MUSE, KMOS, SINFONI), on the ELT (HARMONI), ESI on the Keck telescope, the radio-telescopes ALMA, SKA, LOFAR, MWA but also space-missions like ATHENA and

⁴https://www.eso.org/sci/meetings/2015/Rainbows2015/Talk_Files/DAY1/szalay-eso-rainbows-2015.pdf

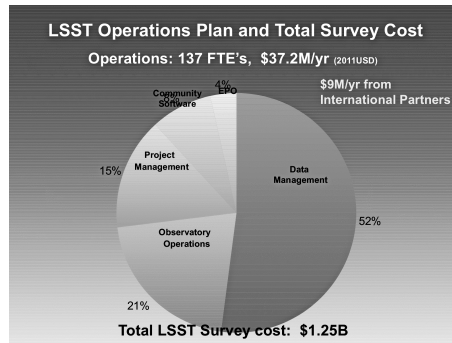


Figure 8. Distribution of the operations budget for the LSST. More than half of that operations budget is spent on data-management. (From "Large Synoptic Survey Telescope", Tony Tyson, Oxford, 16.09.2013)

JWST (MIRI, NIRSPec) will provide 3D data-cubes. At this moment in time, proper source-extraction, source-classification, analysis and visualization tools are only at the verge of being developed.

One such tool, that also provides first scientific analysis, is the ALMA Data Mining Toolkit (ADMIT) (Teuben et al. 2015). After source-finding in 3D ALMA data-cubes, ADMIT also extracts spectra along the third dimension, finds spectral lines and uses the provided redshift information to attempt line-identification. In Fig. 9 ADMIT has automatically detected Hydrogen cyanide in an ALMA observation of Titan.

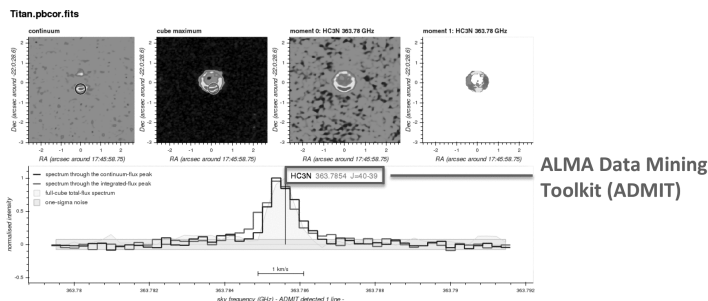


Figure 9. Beta version of an ALMA preview showing line-identification using the ALMA Data Mining Toolkit (ADMIT)

3. Machine Learning

It is anticipated that in astronomy the transition to automated first-analysis will need to be done using Machine Learning methods. Those methods may be needed in the future even in the data-processing process itself, but certainly will play an important role in the quality assurance processes, the source-extraction and the source-classification processes. All those are closely related to the capabilities of Science Archives. Has

machine-learning played only a very minor role in astronomy in the past, maybe with the exception of the the source extraction software *sextractor* (Bertin & Arnouts 1996), the situation has dramatically changed very recently with an explosion of publications on Artificial Intelligence in astronomy. 23 articles have been appearing on the ArXiv preprint server in a period as short as two months.

While often astronomy leads developments that are then later taken up in other scientific fields (e.g. VO, FITS standard), for Machine Learning, other scientific disciplines are ahead. For example the largest scientific calculation carried out to date was a Machine Learning application of iterative Random Forests with 2.36 Exaops carried out at the Oak Ridge National Laboratory, TN, USA, by the group around Daniel Jacobsen. Astronomy can move ahead fast, learning from the findings of other disciplines.

4. Catalog of the Universe

Once advanced Machine Learning techniques are available, we propose that the astronomical community starts undertaking an effort to build the "Catalog of the Universe" (CU) trying to link and classify as many observations from as many facilities as possible into a single ultimate master catalog.

The unprecedented catalogs of GAIA and - soon - LSST, once cross-matched against each-other, can be used to set the reference. Existing catalogs, e.g. the more than 18000 catalogs in Vizier can then be cross-matched against that reference. Machine Learning techniques - with the help of existing spectral information, as well as redshift information and potentially even with the help of Spectral Energy Distribution modeling for the various object types - can be employed to assign probabilities to the possible matches found, based on the matches that have already been identified in the catalog.

Future Machine Learning methods then can be used to extract complex sources from the original data (lensing, arc-finding, jet-finding, ...) and can be run on the original data of the major facilities and instruments. Higher-level information can be extracted, and counterparts of catalog objects in the CU can be identified. Those might not necessarily appear in the facility's catalog as typically very conservative sigma-cuts are applied.

With SIMBAD and NED the astronomical community already owns a treasure-trove of classifications and links between objects, all manually verified. Those catalogs can be used to train the cross-match of the other catalogs and serve as a training set for supervised classification of the sources in the CU.

Such a master catalog would be a quantum leap for astronomy. It would be fully query-able and could be used for multi-wavelength astronomy, but also for outlier detection, statistics or even to drive a set of robotic telescopes carrying out observations to resolve the most interesting apparent conflicts in the catalog.

5. User Experience

The driver behind the described efforts is to improve the User-Experience ("how it feels") for PIs and archival researchers. Typically observatories are built "bottom-up": After the general science capabilities are specified, a conceptual design is carried out, then detailed designs, design reviews, construction and software implementation fol-

low. The details of the actual User-Experience often are designed at the very end of the process. Comparing this to the mobile phone industry, telescopes are built like e.g. the Nokia 6263 from 2007: The phone could very well be used, but nevertheless did not have a User-Experience remotely comparable to the one offered by the first Apple iPhone in the same year, despite having had the same features.

We argue that the observatories as a whole - including the process of the generation of astronomical data-products as well as of archives - should take advantage of existing modern user-centric design principles like DesignThinking⁵.

6. Conclusions

Astronomy is a discipline of science where the reduced data-products are comparably simple due to the fact that the objects of study, photons, are simple particles. Over the last decades a number of best practices for astronomical archives serving those data-products have emerged. With the exponential increase of data taken, however, large challenges are ahead for observatories requiring to process data to higher levels to keep the amount still manageable for analysis. Artificial Intelligence here seems to be able to come to the rescue and may even allow the astronomical community to build the Catalog of the Universe. In order to allow PIs and archival researchers to most effectively convert the data-products and analysis into scientific result, we argue to embrace user-centric design principles.

References

- Bertin, E., & Arnouts, S. 1996, *A&AS*, 117, 393
- Boch, T., & Fernique, P. 2014, in *Astronomical Data Analysis Software and Systems XXIII*, edited by N. Manset, & P. Forshay, vol. 485 of *Astronomical Society of the Pacific Conference Series*, 277
- Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. 2000, *A&AS*, 143, 33
- Fernique, P., Allen, M. G., Boch, T., Oberto, A., Pineau, F.-X., Durand, D., Bot, C., Cambrésy, L., Derriere, S., Genova, F., & Bonnarel, F. 2015, *A&A*, 578, A114. 1505.02291
- Stoehr, F. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shorridge, & R. Wayth, vol. 512 of *Astronomical Society of the Pacific Conference Series*, 511
- Stoehr, F., Grothkopf, U., Meakins, S., Bishop, M., Uchida, A., Testi, L., Iono, D., Tatematsu, K., & Wootten, A. 2015a, *The Messenger*, 162, 30. 1601.04499
- Stoehr, F., Lacy, M., Leon, S., Muller, E., & Kawamura, A. 2015b, in *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*, edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of *Astronomical Society of the Pacific Conference Series*, 69
- Stoehr, F., Manning, A., Moins, C., Jenkins, D., Lacy, M., Leon, S., Muller, E., Nakanishi, K., Matthews, B., Gaudet, S., Murphy, E., Ashitagawa, K., & Kawamura, A. 2017, *The Messenger*, 167, 2
- Teuben, P., Pound, M., Mundy, L., Rauch, K., Friedel, D., Looney, L., Xu, L., & Kern, J. 2015, in *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*, edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of *Astronomical Society of the Pacific Conference Series*, 305

⁵open.sap.com/courses/dt1

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Astrocut: A Cutout Service for TESS Full-Frame Image Sets

C. E. Brasseur, Carlita Phillip, Jonathan Hargis, Susan Mullally, Scott Fleming,
Mike Fox, and Arfon Smith

Space Telescope Science Institute, Baltimore, MD 21218, USA

Abstract. The Transiting Exoplanet Survey Satellite (TESS), launched in April 2018, is a planet finding mission much like the Kepler mission. Like Kepler, the TESS data pipeline returns a variety of data products, from light curves and target pixel files to large full frame images. Unlike Kepler, which took full frame images relatively infrequently, TESS takes them at a 30 minute cadence, making the TESS full frame images a large and incredibly valuable scientific dataset. As part of the Mikulski Archive for Space Telescope's (MAST) mission to provide high quality access to astronomical datasets, MAST has built an image cutout service for TESS full frame images. Users can request image cutouts in a variety of ways, and the returned target pixel files are TESS pipeline compatible. We present the use and design of this software, including both the technical considerations and user experience.

1. Introduction

The Transiting Exoplanet Survey Satellite (TESS) was launched on April 18th, 2018. It is equipped with four cameras each with four 2k x 2k CCDs, and a pixel resolution of 21 arcsec/pixel, arranged for a total field-of-view of 24 x 96 degrees. The TESS main mission is planned to last two years, the first surveying the southern hemisphere, the second the northern hemisphere. The mission is divided into 26 sectors each of which lasts for two orbits of the telescope, or about 27 days, and keeps the same pointing throughout.

The TESS pipeline returns a variety of data products, including light curves, target pixel files, and full frame images. Each time TESS takes full a frame image, it produces one image file per CCD, for 16 total. Since it takes them on a 30 minute cadence, over the course of a sector, TESS produces about 800 GB of full frame images. The TESS full frame images form an important scientific dataset, and it will be possible to do time-domain astronomy just on the full frame images. However, full frame images are large and much of the time scientists will need to cut out sections of interest before scientific analysis. With the release of Astrocut, MAST aims to provide a user friendly way to take cutouts from a sector of full frame images, eliminating both the need for scientists to create their own cutout tools, and the need for users to download the entire sector of full frame images when they only need a small section of each image. A secondary goal is to provide users the image cutouts as a TESS pipeline compatible target pixel file so that software that runs on pipeline target pixel files can also be used on Astrocut target pixel files.

There are three components to the Astrocut software stack. Astrocut itself contains the underlying functionality, and is what actually performs the cutouts. It is an open

source python package that is available for users to install themselves. TESScut is a URL-based web service that runs Astrocut on MAST servers so that users do not need to install Astrocut or download full frame images, but instead simply request their desired cutout through the TESScut service, and download only the resulting target pixel file. Lastly, `astroquery.mast.Tesscut` is a python wrapper around the TESScut web service which allows users to request TESS cutouts target pixel files in Python.

2. The Astrocut Software

Astrocut is an open source Python package that contains tools for making cutouts from sets of TESS full frame images. It can be installed through pip, and documentation can be found at <https://astrocut.readthedocs.io>.

```
In [2]: from astrocut import CubeFactory
        ffi_files = glob('data/*fffc.fits')
        cube_file = CubeFactory().make_cube(ffi_files[:10], "cube_3-2.fits", 0)

Completed file 0
Completed file 1
Completed file 2
Completed file 3
Completed file 4
Completed file 5
Completed file 6
Completed file 7
Completed file 8
Completed file 9
Total time elapsed: 2.03 sec
File write time: 0.53 sec

In [3]: from astrocut import CutoutFactory
        cutout_file = CutoutFactory().cube_cut(cube_file, "251.51 32.36", [2,4]*u.arcsec,
        output_path="data", verbose=True)

Cutout center coordinate: 251.51,32.36
xmin,xmax: [28 29]
ymin,ymax: [151 152]
Image cutout cube shape: (10, 1, 1)
Uncertainty cutout cube shape: (10, 1, 1)
Target pixel file: data/cube_3-2_251.51_32.36_1x1_astrocut.fits
Write time: 0.027 sec
Total time: 0.24 sec
```

Figure 1. Using Astrocut to make TESS full-frame image cubes and cutouts.

Astrocut is comprised of two parts, one for making image cube files from sets of full frame images, and one for performing cutouts on the previously created image cube files. Astrocut is designed this way to maximize performance in the cutout functionality at the expense of some up-front work making the cube files.

Prior to implementing the cube file functionality, the best performance we could achieve was ~25 seconds to make a 10x10 cutout over 1348 full frame images, running in parallel with 8 threads. By moving to the cube paradigm we were able to produce the same cutout in about half a second without any parallelization. The way we achieved this speed up was to build a cube file that contains all of the full frame images for a single CCD in one sector. Relevant header information from each individual full frame image is stored in a table and the images themselves are put into one large 3 dimensional image array. When creating the image cube array we transform the axes, putting time on the first axis, which minimizes the seek actions needed when performing a cutout.

The trade off in using these large image cube files is the one-time up front work, which involves investment in both time and memory. It takes about 10 minutes to build an image cube file, and because of the transformation applied to the image array, it must be stored in memory while being built. This means that a system with about 60 GB of memory is required for making an image cube file over a TESS sector. Once the cube file is made however, it does not need to be read into memory again and so can be safely transferred to a less beefy system for making cutouts. To perform a cutout, the user supplies a cube file, RA/Dec, and cutout size to the Astrocut cutout function and is returned a TESS pipeline formatted target pixel file. Figure 1 shows an example of how to use Astrocut, both to create image cubes and cutouts.

3. The TESScut service

The TESScut web service allows access to Astrocut functionality on MAST servers, eliminating the need for users to download full frame images sets themselves. MAST builds a set of image cube files for each sector, and TESScut allows users to request cutouts, which are performed by Astrocut on the cube files and then streamed back to the user.

There are three ways to request cutouts through TESScut. The main TESScut page at <https://mast.stsci.edu/tesscut> includes a web form where users can search by coordinates or target name, and request cutouts if TESS data exists for the specified location. Cutouts can also be requested directly with HTTPS GET requests, within a browser or using any language that allows HTTPS requests to be sent/received. Finally, TESScut can be accessed through the mast Astroquery module. Astroquery is a Python package for querying astronomical web forms and databases (Ginsburg et al. 2017), and MAST maintains a module within it. Figure 2 shows how to request TESS cutouts through the MAST Astroquery module.

```
>>> from astroquery.mast import Tesscut
>>> from astropy.coordinates import SkyCoord
>>> cutout_coord = SkyCoord(107.18696, -70.50919, unit="deg")
>>> hdulist = Tesscut.get_cutouts(cutout_coord, 5)
>>> hdulist[0].info()
Filename: tess-s0001-4-3_107.18696_-70.50919_5x5_astrocub.fits
No.    Name      Ver    Type      Cards  Dimensions  Format
0  PRIMARY      1  PrimaryHDU    45      ()
1  PIXELS       1  BinTableHDU  225    1282R x 12C  [D, E, J, 25J, 25E, 25E, 25E, 25E,
2  APERTURE     1  ImageHDU     134     (5, 5)  float64
```

Figure 2. Using *astroquery.mast* to request TESS full-frame image cutouts.

4. Conclusions

Astrocut is a new Python software package produced by MAST to allow scientists easy access to TESS full frame image cutouts. TESScut is the service layer produced by MAST that lets users request TESS cutouts without needing to download any full frame images themselves. For most users TESScut will be the interface of choice. It is the most straightforward option and allows users to easily request a handful of TESS

full frame image cutouts. Scientists with more specialized needs, for example someone who wants to make a cutout for every star in the TESS field-of-view, or make cutouts that are very large, may find their needs better served by installing Astrocut directly, downloading the full frame image files themselves and making their own cutouts locally. Additionally, because Astrocut is open source, users who want to do something nonstandard also have the option of forking and modifying Astrocut to better suit their needs. Figure 3 summarizes the available interfaces and where to find more information about them.

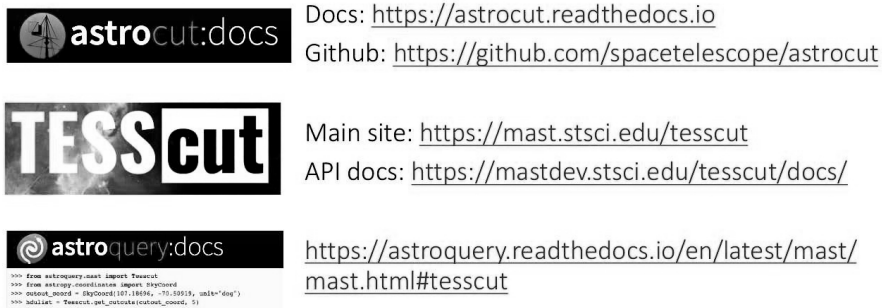


Figure 3. The Astrocut/TESScut ecosystem and where to get more information.

References

Ginsburg, A., Parikh, M., Woillez, J., Groener, A., Liedtke, S., Sipocz, B., Robitaille, T., Deil, C., Svoboda, B., Tollerud, E., Persson, M. V., Séguin-Charbonneau, L., Armstrong, C., Mirocha, J., Droettboom, M., Allen, J., Moolekamp, F., Egeland, R., Singer, L., Barbary, K., Grollier, F., Shiga, D., Moritz Günther, H., Parejko, J., Booker, J., Rol, E., Edward, Miller, A., & Willett, K. 2017, Astroquery: Access to online data resources, Astrophysics Source Code Library. 1708.0004

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

AXS: Making End-User Petascale Analyses Possible, Scalable, and Usable

Petar Zečević,¹ Colin T. Slater,² Mario Jurić,² and Sven Lončarić¹

¹*University of Zagreb, Croatia; petar.zecevic@fer.hr , sven.loncaric@fer.hr*

²*University of Washington, USA; ctslater@uw.edu , mjuric@uw.edu*

Abstract. We introduce AXS (Astronomy eXtensions for Spark), a scalable open-source astronomical data analysis framework built on Apache Spark, a state-of-the-art industry-standard engine for big data processing. In the age when the most challenging questions of the day demand repeated, complex processing of large information-rich tabular datasets, scalable and stable tools that are easy to use by domain practitioners are crucial. Building on capabilities present in Spark, AXS enables querying and analyzing almost arbitrarily large astronomical catalogs using familiar Python/AstroPy concepts, DataFrame APIs, and SQL statements. AXS supports complex analysis workflows with astronomy-specific operations such as spatial selection or on-line cross-matching. Special attention has been given to usability, from conda packaging to enabling ready-to-use cloud deployments. AXS is regularly used within the University of Washington's DIRAC Institute, enabling the analysis of ZTF (Zwicky Transient Facility) and other datasets. As an example, AXS is able to cross-match Gaia DR2 (1.7 billion rows) and SDSS (710 million rows) in 25 seconds, with the data of interest (photometry) being passed to Python routines for further processing. Here, we will present current AXS capabilities, give an overview of future plans, and discuss some implications to analysis of LSST and similarly sized datasets. The long-term goal of AXS is to enable petascale catalog and stream analyses by individual researchers and groups.

1. Introduction

Modern astronomical surveys produce ever larger amounts of data. For example, Large Synoptic Survey Telescope (LSST), which is to start in 2022, will perform approximately 1000 observations of about 20 billion objects (LSST Science Collaboration 2009). This will result in approximately 50 PB of (raw) data over the 10 years of its operations.

Large survey analysis typically involves generating subsets of data by querying the larger catalog (usually with SQL), followed by downloading them (as FITS files, or similar). This is further followed by writing custom (usually Python) scripts to do the bulk of the analysis. While certainly doable, this *subset-download-analyze* workflow can be cumbersome, especially if executed repeatedly. Beyond working on a single catalog, we often want to positionally cross-match objects from two (or more) survey catalogs. This is often solved by pre-computing cross-match tables between catalogs, but as the number of catalogs grows this becomes inefficient ($O(N^2)$, where N is the number of catalogs). Finally, most upcoming servers today will be multi-epoch. There's

therefore a need to enable efficient analyses of *time series*, multiple observations of a single object.

2. Apache Spark as the Workflow Engine for Astronomical Datasets

AXS' goal is to provide a simple, user-friendly, scalable and efficient tool for cross-matching and analyzing data from large astronomical surveys. AXS extends Apache Spark (Zaharia et al. 2016), a general-purpose framework for big data processing. Spark's analytical processing functions are implemented through the Resilient Distributed Dataset (RDD) abstraction. Spark automatically distributes operations on RDDs and executes them in parallel with the minimum involvement of the user.

Spark's rich API can be accessed using SQL, Scala, Java, Python and R interfaces and it offers a breadth of functions: processing of structured and unstructured data, in a streaming or batch fashion, with graph and machine learning algorithms available. Spark can also recover from failures of individual nodes and is efficient and scalable. It enjoys a wide and active user base, both in industry and academic world.

These are the main reasons why we chose Spark to be the base for building AXS upon.

3. Minimal Astronomy-specific Extensions for Spark

Spark already implements a significant fraction of functionality needed to support astronomical data analysis and explorations. We've found that with only two simple extensions – a data partitioning scheme built on existing bucketing support and (completely generic) improvements to the underlying Spark sort-merge join algorithm implementations – it is possible to deliver very performant cross-matching and querying functionality. The purposefully minimal nature of our changes may be more maintainable in the long-run relative to approaches which use astronomy-specific partitioning (e.g., HEALPix-based approaches such as Juric (2011), or the recent effort in Brahem et al. (2018)).

3.1. Data partitioning

Data partitioning scheme employed by AXS is based on the *Zones algorithm* (Gray et al. 2007), a well-known algorithm in the Astronomy community, with adaptations needed to support Spark's distributed architecture. In the AXS data partitioning scheme, sky is partitioned into horizontal bands of fixed height called *zones*. Zone height is one arc-minute by default, which makes 10800 zones. An object's zone is determined based on its *Dec* coordinate using this simple formula (Z is the zone height): $zone = \lfloor (Dec + 90)/Z \rfloor$

These are physically stored by Spark into *buckets*, implemented as Parquet files on a (potentially distributed) filesystem, based on the calculated *zone*: $bucket = zone \% N$ (where N is the number of buckets). The default number of buckets is 500, so there are about 21 zones per bucket, by default.

3.2. Distributed cross-matching

To efficiently cross-match tables bucketed by zones, as previously described, AXS partly relies on Spark's sort-merge join and partly on the contributed epsilon-join (Silva

et al. 2010) optimization. To understand this optimization, consider the following query:

```
select * from gaia, sdss where gaia.zone = sdss.zone AND  
gaia.ra BETWEEN (sdss.ra + DELTA, sdss.ra - DELTA) AND  
distance(gaia.ra, gaia.dec, sdss.ra, sdss.dec) < DELTA;
```

Here, two tables are joined based on the zone column and the distance between two rows is calculated using the provided distance function. Without the range condition on the secondary columns (ra columns), all the pairs of objects from the two tables having the same zone would need to be considered, which can involve a huge number of comparisons. However, Spark is not able to optimize this query and take advantage of the range condition on its own. Epsilon-join optimization uses a moving window which slides over the rows in the right table as the left row changes and hence effectively restricts the number of row pairs considered. This is possible because AXS data is sorted inside buckets by zone and ra columns. The distance function is thus evaluated only for the rows in the moving window, only one pass through the data is needed, and the join process uses the minimal amount of memory.

4. Cross-match performance tests

To benchmark AXS' cross-matching performance we used the Gaia DR2 catalog with 1.7 billion objects and the SDSS catalog with 710 million objects. The cross-matching function calculated the distance between two objects using their RA and Dec coordinates, but an arbitrarily complex function could also be used (e.g., one taking the error-ellipses into account).

The cross-match operation for the benchmarked case results in 227 million rows. The tests included only counting of the resulting rows and no further processing on them was performed. The tests were performed on a single large machine with 512 GB memory, 48 CPUs and fast hard disks. Each Spark executor was given 12 GB of memory and a single CPU core.

We looked at how the cross-matching time depends on the number of Spark executors used. The results are shown in Figure 1. The two different lines in the graph show two sets of tests: with the cold and warm filesystem ("buffer") cache. Each data point on the graph is an average of three tests. The results with the cold buffer cache show the "worst-case" performance of the system which includes reading data from disk. The results with the warm cache more closely represent the time needed by the CPU-limited aspects of the cross-matching operation (we note that the dataset is larger than the memory available, so the data was never fully in the cache).

The lowest times we obtained, visible in the Figure, show that AXS can cross-match these two tables (Gaia DR2 and SDSS) in 25 seconds, with the data cached; and 136 seconds, when the data is not cached. Adding more than 28 executors doesn't offer any substantial performance improvements on the single machine that was used for performance testing.

5. Summary

In this paper we describe and show initial performance benchmarks for Apache eX-tensions for Spark (AXS), a system for performant querying, analyzing, and cross-

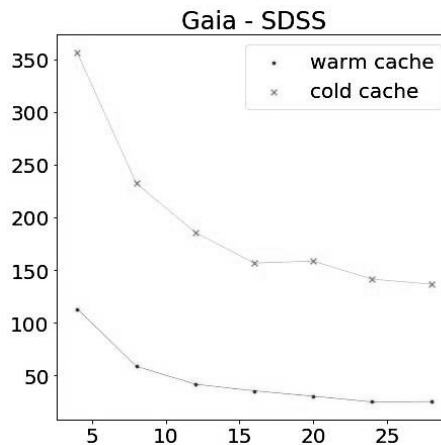


Figure 1. Duration in seconds of cross-matching Gaia DR2 and SDSS catalogs depending on the numbers of Spark executors used. The two curves correspond to tests with data obtained starting with a cold filesystem cache, and tests with some data already residing in the cache. Each data point is an average of three tests.

matching data from astronomical catalogs, built on top of Apache Spark. AXS was created with the goal of making the power of an industry-standard tool such as Apache Spark available to non-specialist astronomers for analysis of large-scale datasets. We next plan to further optimize the AXS Python API and overall performance, as well as to make AXS available both individually and as a part of a cloud-based service.

Acknowledgments. PZ, CTS, and MJ acknowledge support from the University of Washington College of Arts and Sciences, Department of Astronomy, and the DIRAC Institute. University of Washington's DIRAC Institute is supported through generous gifts from the Charles and Lisa Simonyi Fund for Arts and Sciences, and the Washington Research Foundation. MJ acknowledges further support from the Washington Research Foundation Data Science Term Chair fund, and the UW Provost's Initiative in Data-Intensive Discovery.

References

- LSST Science Collaboration 2009, Lsst science book, version 2.0. [arXiv:0912.0201](#)
- Brahem, M., Zeitouni, K., & Yeh, L. 2018, *IEEE Transactions on Big Data*, 1
- Gray, J., A. Nieto-Santisteban, M., & Szalay, A. 2007
- Juric, M. 2011, in *American Astronomical Society Meeting Abstracts #217*, vol. 43 of *Bulletin of the American Astronomical Society*, 433.19
- Silva, Y. N., Aref, W. G., & Ali, M. H. 2010, in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, 892
- Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., & Stoica, I. 2016, *Commun. ACM*, 59, 56. URL <http://doi.acm.org/10.1145/2934664>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

No-SQL Databases: An Efficient Way to Store and Query Heterogeneous Astronomical Data in DACE

Nicolas Buchschacher, Fabien Alesina, and Julien Burnier

University of Geneva, Geneva, Geneva, Switzerland;
nicolas.buchschacher@unige.ch

Abstract. Data production is growing every day in all domains. Astronomy is particularly concerned with the recent instruments. While SQL databases have proven their performances for decades and still performs in many cases, it is sometimes difficult to store, analyze and combine data produced by different instruments which do not necessarily use the same data model. This is where No-SQL databases can help to solve our requirements: how to efficiently store heterogeneous data in a common infrastructure ? SQL database management systems can do a lot of powerful operations like filtering, relation between tables, sub-queries etc. The storage is vertically scalable by adding more rows in the tables but the schema has to be very well defined. In the opposite, No-SQL databases are not restrictive. The scalability is horizontal by adding more shards (nodes) and the different storage engines have been designed to easily modify the structure. This is why it is well suited in the big data era. DACE (Data and Analysis Center for Exoplanets) is a web platform which facilitates data analysis and visualization for the exoplanet research domain. We are collecting a lot of data from different instruments and we regularly need to adapt our database to accept new data sets with different models. We recently decided to opt for NoSQL databases like Cassandra and Solr. This recent change accelerated our queries and we are now ready to accept new data sets from future instruments and combine them with older data to do better science. DACE is funded by the Swiss National Centre of Competence in Research (NCCR) PlanetS project, federating the Swiss expertise in exoplanet research.

1. Introduction

During several years, the DACE database uses a common SQL engine to store the data and the meta data related to exoplanets. Since we collect data from different instruments and research domains, we regularly have to redefine our data models, which is not a good practice in the SQL world. In addition, performances and high availability is a key point for a public service as well as scalability. This could be achieved by adding more powerful hardware with more CPU, memory and disks but with an impact on the price. In the opposite, using NoSQL databases allow us to define very flexible schemas and easily change it. With the recent approaches using parallelism in computer science, we don't necessarily have to buy expensive and powerful machines, but we need a system which has been designed to be fault tolerant and run on cheaper hardware simply by having more cores for redundancy. This is where the NoSQL world enter in the game.

2. Heterogeneous data in a common infrastructure

2.1. Observational data

The DACE database contains some observational data from different instruments and techniques, like spectroscopy (CORALIE, HARPS(-N), ESPRESSO), transit light curves (WASP, TESS, CHEOPS), direct imaging (NACO, PUEO) and astrometry in a near future (HIPPARCOS, GAIA). All these instruments have their own data models with different parameters. Storing all these parameters in a common SQL table is very difficult and would result in having a lot of columns, having some null values which is not a good practice. To solve this, we could have one table for each instrument and do some JOIN queries to collect the data among the entire database. We will see in the next section that some NoSQL databases are typically designed to solve these requirements.

2.2. Synthetic populations: planets formation and evolution

Formation and evolution of the planetary systems is another research domain in which the data are stored and analyzed by the DACE platform. Since the data are synthetic, it is easier to define clear data model and have some relatively fixed schema. The difficulty here is to regularly import billion of data points with a minimal impact on the production system. Having a single node is clearly not possible and we had to think about using a load distributed database management system. In the next section, we describe 2 different databases which solves our requirements both for heterogeneous data and having a distributed load.

3. NoSQL databases: Cassandra and Solr

3.1. Wide column tables

Apache Cassandra (<http://cassandra.apache.org>) is a distributed database developed by the Apache Software Foundation. One of the major difference with a traditional SQL database is the storage structure engine. While a common SQL database stores rows into tables, Cassandra stores columns in key-value pairs. The big advantage of such a design is that having some NULL values in a table doesn't matter and doesn't waste space since the value is simply not present instead of having an explicit NULL. Another feature is the possibility to store thousands of columns in a single table without any big impact on the performances. Finally, adding and deleting some columns in a table is not a complex operation for Cassandra and can be compared to adding or deleting an entry in a dictionary in common programming languages. Figure1 explains how a table is stored in Cassandra.

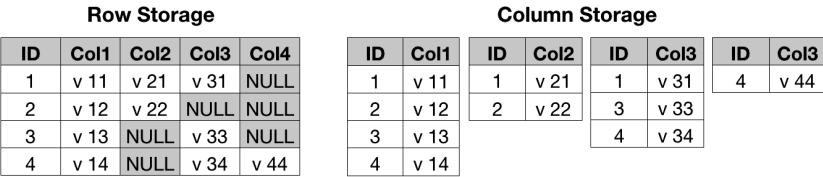


Figure 1. Row vs column oriented storage

3.2. Document database

We saw in the previous section the power of Cassandra to manage a lot of parameters with a flexible schema. Unfortunately, this kind of database is not powerful to execute sophisticated queries like JOINS, filtering and aggregations. Because of the storage design organized in columns, we avoid filtering the data based on another field than the primary key. Even if Cassandra is able to do it, there is a big impact on performances. In the opposite, Solr (<http://lucene.apache.org/solr/>) is a very powerful indexer and search engine based on Lucene (<http://lucene.apache.org>). It is a document oriented database which is able to index a lot of fields and has a big set of search operations, like sort, filtering and counts. One row in a table is equivalent to a document in Solr and each column is equivalent to a field in the document. SolrCloud is the distributed version of Solr and can be compared to Cassandra in the architecture to split and replicate the data across the nodes. The following sub-section describes in details how it works for Cassandra.

3.3. Availability, consistency and partition tolerance: the CAP theorem (Brewer (2000))

Both Cassandra and SolrCloud are distributed databases and consists of having several equivalent nodes working together. Compared to a Master-Slave approach, they are fully distributed, which means that a client can connect to any node and execute a read or write query. The node will automatically forward the request to another node if the data has to be read or write somewhere else and if needed, the result will be merged to finally be returned to the client as if there was a unique server. In addition, it is possible to define a replication factor to ensure high availability in case of a node failure. The Figure 2 shows how the data are split and replicated over the nodes.

The CAP theorem, also known as the Brewer's theorem states that this is impossible for a distributed database to simultaneously provide more than two of the three following requirements:

- **Availability:** every read or write request receives a non-error response
- **Partition tolerance:** the system continues to work properly even if there are some missing messages from nodes
- **Consistency:** every read receives the most recent value or an error

Cassandra clearly satisfies the first two statements and can be classified as an AP system. Availability is satisfied by the fact that all the nodes are equivalents and the client can connect to any of them to send a request. The system is partition tolerant if it has a replication factor > 1 because at least another node will contain the same information and the node from which the query has been executed will automatically switch to the replicated node in case of failure. With such a design, consistency cannot be ensured due to the asynchronous replication of the data across the nodes. But Cassandra implements a mechanism of consistency level for each query. Some basic levels are described below:

- **ONE:** Validation from the closest replica is sufficient to reply.
- **ALL:** All replica have to validate the request for read and writes. If there are some differences between 2 replicas, there is an error.

- **QUORUM:** The majority of the replicas has to validate the request.

There exists a lot of other consistency strategies taking into account the location of the nodes among different datacenter. With this kind of strategy, the client can choose the orientation between very high availability or very high consistency. But because of the CAP theorem, not all the conditions can be satisfied.

4. Challenges and difficulties to use NoSQL databases

We saw in the previous chapters that NoSQL databases like Cassandra and Solr have some big advantages. Not only the schema flexibility but also the distributed concept which is very easy to configure and ensure high availability. But every technology has some drawbacks. A NoSQL system like Cassandra or Solr is not relational, which means that SQL standard queries like JOINS are not available. The merging task has to be done at the user level, in the high level application. In astronomy, using NoSQL databases is not an easy task, since there are a lot of standards like the Virtual Observatory and its TAP (Table Access Protocol) which is strongly based on SQL syntax. But from computer science point of view, this new generation of databases offer a lot of flexibility and some powerful tools in the big data era.

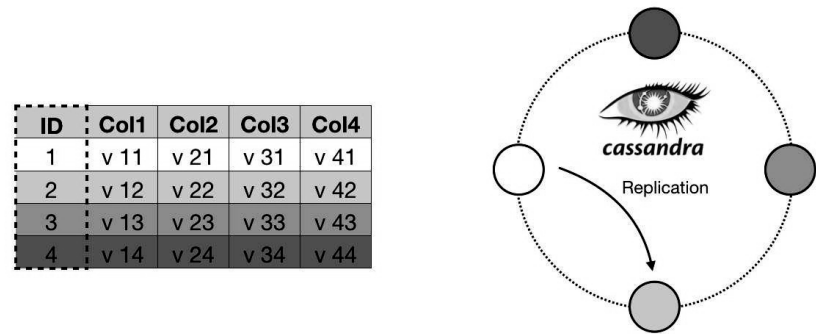


Figure 2. Table storage in Cassandra based on partition key

Acknowledgments. This work has been carried out within the framework of the National Centre for Competence in Research PlanetS supported by the Swiss National Science Foundation. The authors acknowledge financial support from the SNSF. This publication makes use of DACE, a Data Analysis Center for Exoplanets, a platform of the Swiss National Centre of Competence in Research (NCCR) PlanetS, based at the University of Geneva (CH).

References

Brewer, E. 2000, Towards robust distributed system. Symposium on Principles of Distributed Computing (PODC), URL <https://people.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Data-driven Space Science at ESAC Science Data Centre

Beatriz Martinez,¹ Isa Barbarisi,² Juan Gonzalez,² Monica Fernandez,¹
Caroline Laantee,³ Bruno Merin,³ Sara Nieto,¹ Hector Perez,¹ Jesus Salgado,⁴
and Pilar de Teodoro⁵

¹*ESAC Science Data Centre, RHEA Systems S.A for ESA, Madrid, Spain;*
beatriz.martinez@esa.int

²*ESAC Science Data Centre, SERCO for ESA, Madrid, Spain*

³*ESAC Science Data Centre, ESA, Madrid, Spain*

⁴*ESAC Science Data Centre, QUASAR for ESA, Madrid, Spain*

⁵*ESAC Science Data Centre, AURORA for ESA, Madrid, Spain*

Abstract. For many scientists nowadays, the first step in doing science is exploring the data computationally. New approaches to data-driven science are needed due to the big increase of space science mission's data in volume, heterogeneity, velocity and complexity. This applies to ESA space science missions, whose archives are hosted at the ESAC Science Data Centre (ESDC). Some examples are: Gaia archive, whose size is estimated to grow up to 1PB and 6000 billion of objects, Solar Orbiter archive, which is expected to handle several time series with more than 500 millions of records, and Euclid archive, which shall be able to handle up to 10PB of data. The ESDC aims, as a major objective, to maximize the scientific exploitation of the archived data. Challenges are not limited to manage the large volume of data, but also to allow collaboration between scientists, to provide tools for exploring and mining the data, to integrate data (the value of data explodes when it can be linked with other data), or to manage data in context (track provenance, handle uncertainty and error). In this paper, those solutions, which ESDC is exploring in different areas for handling these challenges, will be presented. Specifically: storage of big catalogues through distributed databases (e.g., Greenplum, Postgres-XL); storage of long time series in high resolution via time series oriented databases (TimeScaleDB); fulfill data analysis requirements via Elasticsearch or Spark/Hadoop; and enabling scientific collaboration and closer access to data via JupyterLab, Python client libraries, and integration with pipelines using containers.

1. Introduction

The science data from ESA space science missions are archived at the ESAC Science Data Centre (<http://archives.esac.esa.int>). ESDC has the responsibility to ensure that the data hosted in its archives are of the highest quality, scientifically validated and with all the necessary services, documentation and tools to maximize its scientific exploitation. Data from more than 20 space science missions are archived at ESDC, covering the fields of astrophysics, like Herschel or Gaia missions; planetary, like Mars Express or Rosetta; and heliophysics like Cluster or SOHO missions.

2. Archives Data Storage

In the graph below (Figure 1, left) the evolution in time of our data repository with the different missions is presented. At the time of writing this paper Gaia archive (GACS, Salgado et al. (2017)) has the biggest repository, with around 1.2 PB, but is expected that the Euclid Science Archive will add 10PB. Important to note that it is not expected that all data produced by Euclid mission will reach the science archive. Focusing on database sizes (Figure 1, right), it is remarkable the evolution in size among different missions going from small databases to big ones. Again GACS is the largest with 20 terabytes, and this number will increase much more with its data release 3 (DR3).

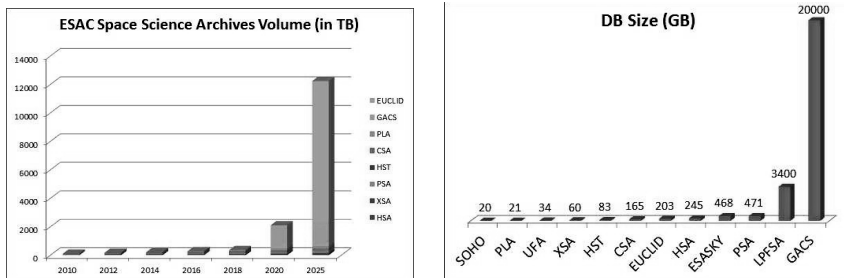


Figure 1. *Left:* Evolution of archives repositories. *Right:* Database size per mission archive

The wave of big data has arrived to the space science missions, producing large catalogs and huge file repositories, and the figures increase every year. But ESDC challenges do not restrict to manage large volume of data and high heterogeneity. The archives also need to provide environments where the scientists can work together, share easily their results, etc. That is, they must help scientists to leverage the exploitation of data.

3. Solutions Adopted and Implemented

Within ESDC, engineers and scientists work together very closely. This allows to provide technical solutions based on scientific use cases. Below, an outline of how some of the challenges are being addressed currently.

- Scientific collaboration and code to data through VO protocols.** The VO protocols are part of the ESDC infrastructure. A first analysis directly on the data is enabled with the IVOA TAP protocol, since it specifies how to interact with a big catalog by using ADQL. As already implemented in the Gaia archive, scientists can share their work/results either via VOSpace (kind of a dropbox for the VO) that allows users to upload content and share it with colleagues (either public or privately). Or by the provision of private areas to registered users, where they can upload their own tables and later on, share it with other users, use them to perform crossmatches, and so on.
- Handling of large datasets in RDBMS.** A typical approach is to reduce the data to manage. The Lisa Path-Finder Science Archive (LPFSA) is an exam-

ple of this in two different areas. At database level, the table partitioning technique provided from Pg10+ is used to solve the storage and query performance issue of tables with more than 10 billion rows. At human visualization level, the “Largest-Triangle-Three-Buckets” algorithm (Steinarsson 2013) was selected and validated scientifically to downsample the amount of data for interactive visualization of more than 2 million points, without degrading the shape of the resulting plot.

- **Exploring heterogenous data.** ESASky (López Martí et al. 2017) is a science-driven discovery portal for most of astronomical missions. Users can explore the the sky in multiple wavelengths, quickly identified the data available for their targets and retrieve the relevant science products for the corresponding archives.

4. Solutions in Evaluation or Under Prototyping

4.1. Massive Parallel Processing (MPP) for Big Catalogues

Extra large catalogues, like Euclid will have, require moving away from traditional databases architecture. ESDC is evaluating the possibility of handling them through MPP with distributed relational databases (de Teodoro et al. 2017). Specifically with databases that scale-out PostgreSQL. PostgreSQL is an open source system, with a big support community and with key advantages as high availability, fault tolerance and extensions like spherical queries, healpix, q3c or postgis. Three main options exist in the market for distributed PostgreSQL architecture: Postgres-XL, CitusData and GreenPlum. The continuity of Postgres-XL project is not clear, for this reason, even when the Gaia archive uses Postgres-XL (González-Núñez et al. 2017), ESDC is exploring also the possibility of using Greenplum or Citus. Benchmarks will be done with a typical Euclid query profile, so the results can be compared with othe MPP techniques.

Following with Massive Parallel Processing, a prototype based on a Spark cluster has been created. The purpose of this prototype is to demonstrate if and how a large Euclid dataset could be analysed with Spark. To support Spark to scale-out, Kubernetes will be added to the prototype. Re-using the Euclid query profile, benchmarks will be done and the results compared with the ones obtained with distributed databases.

4.2. Specific Searches by Data Nature

In situ instruments of Solar Orbiter mission (launch 2020), when in high resolution mode, will produce large time series datasets. Timescale, an open source time series DB with a PostgreSQL kernel, is being tested at ESDC. In Timescale the primary key is the time and the partitioning is time-based. Those facts make it more scalable and easy to administrate than regular relational databases. The drawback is the big size of the indexes required to make really fast queries for Solar Orbiter archive uses cases. For this reason, other non-SQL time series databases as OpenTSdb or InfluxDb will be evaluated.

A different use case can be found in the Planetary Science Archive (PSA). There a prototype to perform full text search over an heterogeneous set of data was implemented. For tests, files in PDS4 format of Exomars16 were selected. The prototype demonstrated that the solution was feasible and in the future it will become one of the PSA functionalities.

4.3. Code to the Data and Scientific Collaboration

Focusing on the scientists work, nowadays the tendency is to use Python to analyse and visualize data. As first step towards the integration of data processing in our archives, several Python libraries that provide access to our archives are being implemented. Already available in Astropy is the GAIA module, and soon an ESASky module (pyESASky), and a Hubble module will join it.

In the future, scientists will be able to exploit ESDC archives data through the Science Exploitation and Preservation Platform (SEPP). In the meantime, a prototype of a JupyterHub environment at the archives has been put in place. Currently, some ESA internal scientists can create their Jupyter Notebooks to access archives data, analyze and visualize it. There is an on-going study to scale-out the data analysis capacity of these Jupyter Notebooks with Spark using 'pySpark' library.

Moving towards the "code to the data" paradigm requires the ESDC VOSpace storage to scale-out. As a solution, "Ceph" is under study. Ceph is a software defined storage solution with great advantages: massively scalable (to Exa-Bytes), highly reliable, easy to manage, and open source.

5. Conclusions

ESDC's objective is to avoid the scientist drowning in the wave of big data. But there is not a unique solution, thus a variety of solutions (based in science use cases) are proposed in different areas:

- storage of big catalogues through distributed databases,
- storage of long time series in high resolution via time series oriented databases,
- data search and processing via specialized analysis engines,
- and enabling scientific collaboration and closer access to data via JupyterLab, Python client libraries and integration with pipelines using containers.

References

- de Teodoro, P., Nieto, S., Salgado, J., & Arviset, C. 2017, in Proceedings of the 2017 conference on Big Data from Space, 193
- González-Núñez, J., Gutiérrez-Sánchez, R., Salgado, J., Segovia, J. C., Merín, B., & Aguado-Agelet, F. 2017, *Astronomy and Computing*, 20, 77
- López Martí, B., Merín, B., Giordano, F., Baines, D., Racero, E., Salgado, J., Henar Sarmiento, M., Gutiérrez, R., de Teodoro, P., González, J., Segovia, J. C., Nieto, S., Norman, H., & Arviset, C. 2017, in *Highlights on Spanish Astrophysics IX*, 660
- Salgado, J., González-Núñez, J., Gutiérrez-Sánchez, R., Segovia, J. C., Durán, J., Hernández, J. L., & Arviset, C. 2017, *Astronomy and Computing*, 21, 22. 1710.10509
- Steinarsson, S. 2013, Master's thesis, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Bringing Together the Australian Sky - Coordination and Interoperability Challenges of the All-Sky Virtual Observatory

Simon O'Toole and Katrina Sealey

AAO-MQ, Macquarie University, NSW, Australia; simon.otoole@mq.edu.au

Abstract. The Australian All-Sky Virtual Observatory (ASVO) consists currently of 5 nodes. There are 2 nodes with optical astronomical data; Data Central (Macquarie University) and SkyMapper (Australian National University). There are 2 nodes with radio data; Murchison Wide Field Array (MWA, Curtin University) and CSIRO ASKAP Science Data Archive (CASDA, CSIRO). The last node is the Theoretical Astrophysical Observatory (TAO, Swinburne University). These 5 nodes work together under the unified ASVO.

The Australian astronomical user community is driving multi-node and multi-wavelength use cases, for example, querying Data Central spectroscopic data with SkyMapper imaging data. Meeting the user requirements of the community comes with complexities and challenges. Some of the challenges we are facing include a single sign-on (unified authorisation/authentication) and the querying and representation of very different remote data, such as, overlaying Galactic and Extragalactic All-sky MWA Survey (GLEAM) data stored in Western Australia with imaging data stored in Eastern Australian states. This presentation will discuss the challenges and successes in both co-ordinating the Australian ASVO and providing interoperability across the 5 nodes.

1. Introduction

Australia has a long tradition of online data access for both raw data archives and science data products, including surveys such as the 2dF Galaxy Redshift Survey. The original Aus-VO was a founding member of the IVOA, however the project faltered due to a lack of funding and immature technologies. A new Australian virtual observatory was launched in 2013 with the All-Sky Virtual Observatory (ASVO). It consists of five nodes covering different aspects of astronomy; these are outlined below. Following this is a discussion of some of the challenges faced by the ASVO and some of the successes.

2. The ASVO Nodes

Along with the features of each of the ASVO nodes described below, they also serve data through IVOA-compliant services such as TAP and Simple Image Access where appropriate.

2.1. MWA

The MWA node serves pre-processed uncalibrated data from the Murchison Widefield Array (see <https://asvo.mwatelescope.org/>). The telescope and ASVO node are operated out of Curtin University and the data are hosted at the Pawsey Supercomputing Centre. The MWA is a low frequency radio telescope operating between 80 and 300MHz and is one of the two Australian Square Kilometre Array (SKA) precursors. It has been operating since 2013 and there are currently 28 petabytes of publicly available data. The ASVO node averages data into smaller volumes with the aim of enabling access for astronomers not directly involved in the project.

2.2. SkyMapper

The SkyMapper node (see <http://skymapper.anu.edu.au/>) serves process and calibrated multi-epoch, multi-band images and photometry from the SkyMapper Southern Sky Survey. The data are taken with a specially built 1.3m telescope located at Siding Spring Observatory, and operated by the Australian National University. There is currently one petabyte of data stored in the node, with the first data release in 2016 and a second due in 2019.

2.3. CASDA

CASDA is the CSIRO ASKAP Science Data Archive and serves data from the Australian SKA Pathfinder telescope (see <https://casda.csiro.au/>). ASKAP is a 36 antenna radio telescope and is the second of two Australian SKA precursors. CASDA hosts science-ready data products and in full telescope operational mode will collect 5 petabytes per year. The first data release through CASDA was in late 2015.

2.4. TAO

The Theoretical Astrophysical Observatory (TAO) was launched in March 2014 and was the first of the new ASVO nodes (see <https://tao.asvo.org.au/tao/>). It hosts cosmological and galaxy formation simulations for astronomers and allows them to build virtual universes based on semi-analytic models. Since it was launched over 1000 of these virtual universes has been built.

2.5. Data Central

Data Central was originally set up by the Australian Astronomical Observatory to host raw Anglo-Australian Telescope (AAT) data, as well as survey data products (see <https://datacentral.org.au/>). It holds 45 years worth of AAT data, along with survey data releases from e.g. the GAMA, SAMI and GALAH surveys. Data Central has a web UI and IVOA services, and a REST API will be released soon.

3. Working as One

3.1. Challenges

The biggest challenge that five nodes of the ASVO face is that it is expected that they will act as one unified virtual observatory. This is difficult, since each node has different infrastructure, requirements, user management, and politics.

In 2017, a review of the ASVO by Astronomy Australia Limited (the management agency for public funding) produced a series of recommendations. These included that the nodes should seamlessly integrate, so that a user would not know which node their data was coming from, simply that it came from the ASVO. The nodes each use IVOA protocols, which goes a long way to addressing this requirement. However the biggest obstacle is access control: how can we connect the five nodes, each with their separate user management systems? Only by solving this problem can we truly claim that the ASVO is seamlessly interoperable.

Another requirement of the ASVO Review was to adopt and implement the FAIR principles (Findable, Accessible, Interoperable, Reusable). While most of the principles (set out at <https://www.force11.org/>) are met by the nodes individually, another big challenge is to make the ASVO FAIR as a whole. Again, the IVOA protocols and standards will be a key aid in this task, but access control is still the main problem to solve.

3.2. Successes

These challenges are great, but we have had some successes. Since the ASVO nodes are spread right across Australia, in order to break down the distance, we hold monthly technical meetings, and biannual retreats that alternate between the west and east coasts of the country.

We are currently trialling on-the-fly cross-matching of data held by Data Central with that held by SkyMapper. This is made possible by Data Central's use of PrestoDB as a query engine; PrestoDB can query most flavours of SQL (as well as some noSQL) and we have written an ADQL plugin for it to allow cone-search queries. We are also building a set of shared tools, including a spectrum viewer (SkyMapper/Data Central), the CASDA VO tools (CASDA/MWA) and pyvospace (made by MWA but shared to all).

3.3. The Future

Finally, we have started work on unifying access control, by implementing an OAuth2.0 and Open ID Connect system. ORY/Hydra (<https://www.ory.sh>) was identified as a lightweight Identity Provider (IDP) system, which allows straightforward integration with an existing local IDP. We will roll this out across the ASVO in 2019. One of the outstanding questions for this system is: can we make it IVOA compliant?

4. Summary

In many ways, the ASVO can be viewed as a miniature version of the international virtual observatory community: independent systems trying to develop and maintain interoperability with each other to enable astronomers to do great research. The key aspects of our success so far have been working together collaboratively, rather than competitively; sharing knowledge across all five nodes; and an open data access policy.



ADASS breakfast (Photo: Unknown)



Sébastien Derriere giving an all-sky astronomy tutorial (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Driving Gaia Science from the ESA Archive: DR2 to DR3

J. González-Núñez,^{1, 2} J. Salgado,¹ R. Gutiérrez-Sánchez,¹ J.C. Segovia,¹ J. Durán,¹ E. Racero,¹ J. Osinde,¹ P. de Teodoro,¹ A. Mora,¹ J. Bakker,¹ U.Lammers,¹ B. Merín,¹ C. Arviset,¹ F. Aguado-Agelet²

¹*European Space Astronomy Center (ESAC), Madrid, Spain*
 juan.gonzalez.nunez@esa.int

²*ETSE Telecomunicación, Universidade de Vigo, Campus*
Lagoas-Marcosende, 36310 Vigo, Spain

Abstract. Released 25th April, Gaia DR2 hosted in the ESA Gaia archive is leading a paradigm shift in the way astronomers access and process astronomical data in ESA archives.

An unprecedented active community of thousands of scientists is making use of the latest IVOA protocols and services (TAP, DataLink) in this archive, benefitting from remote execution and persistent, authenticated, server side services to speed up data exploration and analysis. The availability of a dedicated Python library for this purpose is connecting the archive data to new data processing workflows.

The infrastructure serving this data has been upgraded from DR1, now making use of replication, clustering, high performance hardware, and scalable data distribution systems in new ways for ESA astronomical archives. VO orientation of the archive has been strengthened by the provision of Time Series in DR2 through use of a VO-aware format and protocol.

1. DR1 to DR3: The Challenges

Exposing Gaia data to the scientific community generates several grand challenges, based in the complexity of the data generated, as well of the data volume. Not only scalability plays a key role, but also the overall system performance in order to deal with an increasing volume of scientific questions that are required for data indexing and large scale analysis.

2. Gaia CU9 and the ESA Gaia Archive

Development of the ESA Gaia Archive (Salgado et al. 2017) is integrated within the wider Coordination Unit 9 for the Data Processing and Analysis Consortium for the Gaia mission (DPAC). Responsibilities are split in work packages that cover main activities, that revert into the final Archive: Visualization, Validation, Operations, etc. The ESA Gaia Archive, developed in the ESAC Science Data Centre (ESDC) (Arviset 2015), plays the role as the central repository for the archived data, and provides the necessary interfaces to the scientific community, as well to the relevant stakeholders within the coordination unit.

The utilization fully Open APIs is fundamental for the success in this central role. All of the server side capabilities (APIs) required for the Archive user interface functionality are exposed to the general public and documented, even through the development of specific libraries in languages like Java and Python. Virtual Observatory protocols like TAP, UWS, DataLink, VOSpace are the core backbone of the Gaia Archive server side, not an on-top addition over tailored protocols. In addition, when a VO protocol does not fully fit the purpose, it is extended, keeping compatibility with existing tools and services, and bringing useful feedback to the VO for protocol evolution.

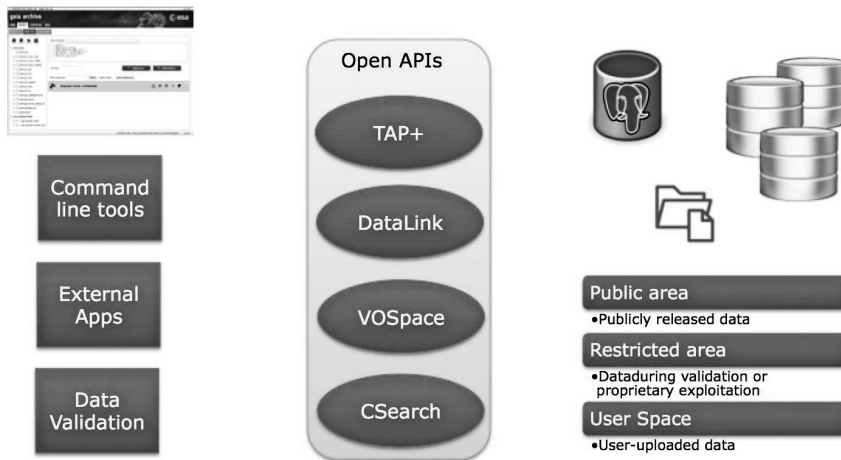


Figure 1. Gaia Archive Open APIs as of DR2

3. Data Release 2: Scaling Up

As of Data Release 2 (Gaia Collaboration et al. 2018), main architectural advancement is focused in the provision of the associated dataset of epoch photometry. The implementation included does not only provide an effective way to query and retrieve this dataset, but also introduces a fully scalable architecture for the future inclusion of larger product datasets as of Data Release 3 and 4.

The architecture is based in a system optimization, where the TAP+ interface hosts catalogues, source classification, and SSOs. Hosting in this system efficiently “indexable” data provides all the benefits from storage in relational databases behind for exposure in the TAP+ interface.

The second part of the data storage now happens through the DataLink interface included for this release, where associated data products are stored. DataLink allows for efficient DataModel-agnostic search over large datasets based on product level meta-data, what makes it the perfect fit for querying Gaia spectra or light curves.



Figure 2. Data split as of Data Release 2 and associated technologies

4. Towards Data Release 3

In the development of DR3, tackling the adequate provision of tools for the scientists to efficiently query and analyze the increasing data volumes, in particular of associated data products, constitutes the main challenge. Notwithstanding this, other challenges are faced as the richness of the associated data products set is also increased, with newer types of products in the generation pipeline, including a large volume of Spectra.

4.1. Data Analysis Trending Tools

A very light inspection of 4 months of usage logs (July-October 2018) provides many hints about the way astronomers are increasingly accessing the ESA Archive.

- Access to classic simple VO protocols like Cone Search remains high. With only 1 month of data since its release, it constitutes 54% of the queries received for a 4-month period.
- ESDC contributed Python library `astropy.gaia` constitutes almost the same volume of batch analysis queries received than the aggregation of all other methods of programatic access to the archive (shell access, other third party libraries, VO tools, etc.)
- Java library remains very lightly used, only for niche applications, with only 540 queries submitted for this 4 month period

The second point is very clearly reflected in Fig. 3, showing an almost exponential query growth through the ESDC contributed Python library to Astroquery library within the Astropy project.

This increasing interest of the community is directly reflected in the ESAC Science Data Centre planned contributions to open libraries, including:

- Provision of DataLink capabilities to `astropy.gaia`.

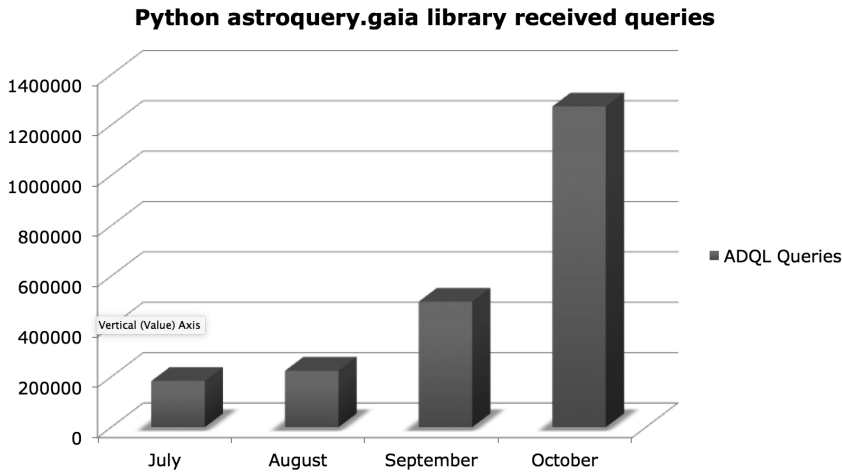


Figure 3. Volume of queries received for users of the astroquery.gaia Astropy/Astroquery module

- ESASky (Giordano et al. 2018) dedicated library (pyESASky), in addition to the already existing data access library (astroquery.esasky), pluggable into any JupyterHub notebook as a widget.
- Python access libraries for other ESA Archives, like the Hubble EHST module

4.2. Code Closer to Gaia Data

Besides the creation of open source libraries to ease the access to and programatic analysis of the larger amounts of data, moving this code closer to the data storage repositories at the main archives becomes a serious productivity increase measure.

ESA efforts are converging into a unified cloud computing platform: the Science Exploitation and Presentation Platform (SEPP). As a pathfinder, a JupyterHub Proof of Concept for SEPP was created by ESDC for JupyterLab awareness workshop, with several demo notebooks made available in the workshop covering different science cases using our platform and libraries.

References

- Arviset, C. 2015, in Science Operations 2015: Science Data Management, id.2, 2
- Gaia Collaboration, Brown, A. G. A., Vallenari, A., Prusti, T., de Bruijne, J. H. J., Babusiaux, C., Bailer-Jones, C. A. L., Biermann, M., Evans, D. W., Eyer, L., & et al. 2018, A&A, 616, A1. 1804.09365
- Giordano, F., Racero, E., Norman, H., Vallés, R., Merín, B., Baines, D., López-Caniego, M., Martí, B. L., de Teodoro, P., Salgado, J., Sarmiento, M. H., Gutiérrez-Sánchez, R., Prieto, R., Lorca, A., Alberola, S., Valtchanov, I., de Marchi, G., Álvarez, R., & Arviset, C. 2018, Astronomy and Computing, 24, 97. 1811.10459
- Salgado, J., González-Núñez, J., Gutiérrez-Sánchez, R., Segovia, J. C., Durán, J., Hernández, J. L., & Arviset, C. 2017, Astronomy and Computing, 21, 22. 1710.10509

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Creating and Managing Very Large HiPS: The Pan-STARRS Case

Thomas Boch¹ and Pierre Fernique²

¹*CNRS, Observatoire de Strasbourg, Strasbourg, France;*
thomas.boch@astro.unistra.fr

²*CNRS, Observatoire de Strasbourg, Strasbourg, France;*
pierre.fernique@astro.unistra.fr

Abstract. HiPS (Hierarchical Progressive Surveys) is a proven Virtual Observatory standard which enables an efficient way to deliver easily potentially huge images collection and allows for fast visualisation, exploration and science applications. CDS has recently published the HiPS for Pan-STARRS g and z-bands images, covering three quarter of the sky at a resolution of 250mas per pixel. We will describe in this paper the challenges we faced and the lessons learned in generating and distributing these HiPS made of 47 million FITS tiles, amounting to 10 trillion pixels and more than 20TB per band. In particular, we will detail the methods we developed to optimize the generation, the storage and the transfer of the HiPS. In addition, a color HiPS, based on the two already available HiPS, has been made available and can be visualized from HiPS clients, like Aladin Desktop (Bonnarel et al. 2000) or Aladin Lite (Boch & Fernique 2014).

Pan-STARRS survey key figures

The Pan-STARRS PS1 survey (Chambers et al. 2016) covers three quarter of the sky in five photometric bands: g, r, i, z and y. Original images come as RICE compressed FITS files whose pixel size is 250 mas. Each band is made up of 200,000 images for a total size of 15 TB.

In the next sections, we will cover the different steps allowing one the creation of the HiPS (Fernique et al. 2015) from images of one Pan-STARRS band and report on our efforts to significantly reduce the overall generation time.

1. Original files download

Downloading of images from the Space Telescope Science with a single wget was not sufficient to take full advantage of the network link between STScI and CDS: the transfer rate was only 12 MB/s. Using several wget connections in parallel increased the transfer rate to 46MB/s in average. Total transfer time for one band went down from 12 to 3.5 days.

2. FITS tiles generation

Pan-STARRS images come as RICE_1 tile-compressed FITS files. In our initial tests, FITS images had first to be uncompressed launching parallel instances of funpack. The Hipsgen tool has been upgraded to support RICE_1 images as input. This improvement saved us 4.5 days and has brought down the FITS tiles generation time to 20 days (on a 5 years old server with 128 GB RAM and 32 hyper-threaded cores)

3. JPEG tiles generation

Generation of JPEG tiles is a 20 days long process and result to generate 6TB of additional HiPS data. We created a Python service, based on the Falcon framework¹ and using astropy.visualization (Astropy Collaboration 2018) library for the heavy-lifting, which generates on-the-fly JPEG tiles from the existing FITS tiles. Thanks to Apache rewriting rules, access to those JPEG tiles is totally transparent for HiPS clients.

The service generating the JPEG tiles has been extended to allow for user-defined choosing of pixel cuts, stretch and color map, as demonstrated in figure 1 . It can be tested at

<http://aladin.unistra.fr/AladinLite/showcase/dynamic-tiles-generation/>.

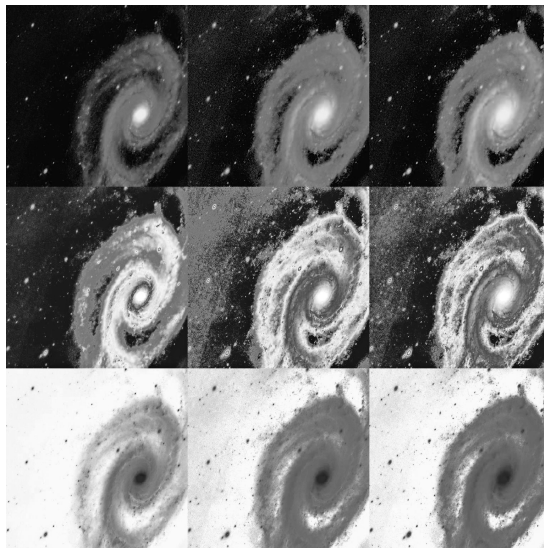


Figure 1. PanSTARRS z HiPS tile (order 9/index 705732) generated with different color maps (from top to bottom: viridis, terrain and Reds) and stretch functions (from left to right: linear, sqrt and asinh). This illustrates the flexibility of our service converting FITS tiles to JPEG.

¹<https://falconframework.org/>

4. Transfer to production server

Our initial transfer process based on rsync solely was slow, averaging 12MB/s (1TB/day) on an internal Gigabit link. Using parsync (a parallel rsync wrapper²), we increased the transfer rate to 100 MB/s. The transfer time has been reduced from 20 days to less than 3 days.

5. RGB color tiles generation

A RGB color HiPS has been generated, the red channel being assigned to the z band, the blue channel to the g band and the green channel being the mean between z and g bands pixel values.

The tiles have been generated with a Python script, reading the existing FITS tiles and outputting JPEG tiles using a with Lupton-like *arcsinh* stretch to maximize contrast. Different cut parameters have been applied according to the tile orders, as to optimize contrast and large and small scales.

The figure 2 above is this generated color HiPS visualized in Aladin Desktop.

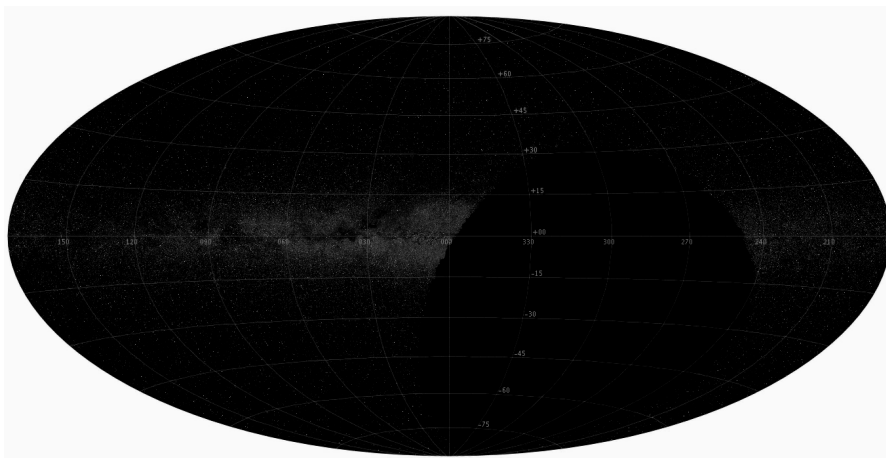


Figure 2. Color HiPS for PanSTARRS data, generated from z and g bands

Pan-STARRS HiPS key figures

As of today, three Pan-STARRS bands (g, z and y) are available as HiPS, described and available from the HiPS network.

Each HiPS has been created with a resolution of 200 mas (HEALPix order 20), slightly oversampled from the original 250 mas resolution. This represents 10 trillion pixels per band, divided into 47 million FITS tiles, and amounting to 25 TB.

²<http://moo.nac.uci.edu/hjm/parsync/>

Conclusion and perspectives

Step	Initial duration	Improvement	New duration
Images download	12 days	parallel wget	3.5 days
FITS tiles generation	25 days	RICE_1 support	20 days
JPEG tiles generation	20 days	On-the-fly generation	0
Transfer to production server	20 days	parsync (parallel rsync)	3 days
Total	77 days		26.5 days

As summarized in table 5, the total HiPS generation time (from original FITS download to release) of one Pan-STARRS band has been reduced from 80 to 30 days. This new streamlined process has been put into practice for the creation of the z band now available in the HiPS network and will be applied to the generation of the remaining r, i and y bands, but also to other large image surveys we will process in the future. It also provides us with additional flexibility regarding the creation of JPEG tiles. Creation of HiPS tiles on the server-side leads the way to some creative science-driven usage of the HiPS standard with minimal changes to the client applications. We are currently developing a prototype allowing one to generate on-the-fly color HiPS times from three or more existing HiPS surveys.

References

Astropy Collaboration 2018, AJ, 156, 123. 1801.02634

Boch, T., & Fernique, P. 2014, in Astronomical Data Analysis Software and Systems XXIII, edited by N. Manset, & P. Forshay, vol. 485 of Astronomical Society of the Pacific Conference Series, 277

Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. 2000, Astronomy and Astrophysics Supplement Series, 143, 33

Chambers, K. C., Magnier, E. A., Metcalfe, N., Flewelling, H. A., Huber, M. E., Waters, C. Z., Denneau, L., Draper, P. W., Farrow, D., Finkbeiner, D. P., Holmberg, C., Koppenhoefer, J., Price, P. A., Saglia, R. P., Schlafly, E. F., Smartt, S. J., Sweeney, W., Wainscoat, R. J., Burgett, W. S., Grav, T., Heasley, J. N., Hodapp, K. W., Jedicke, R., Kaiser, N., Kudritzki, R. P., Luppino, G. A., Lupton, R. H., Monet, D. G., Morgan, J. S., Onaka, P. M., Stubbs, C. W., et al. 2016, ArXiv e-prints, arXiv:1612.05560. 1612.05560

Fernique, P., Allen, M. G., Boch, T., Oberto, A., Pineau, F. X., Durand, D., Bot, C., Cambrésy, L., Derriere, S., Genova, F., & Bonnarel, F. 2015, A&A, 578, A114

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Archive-2.0: Metadata and Data Synchronization Between MAST, CADC, and ESAC

Patrick Dowler,¹ Maria Arevalo,² Adrian Damian,¹ Javier Duran,²
Daniel Durand,¹ Severin Gaudet,¹ Jonathan Hargis,³ Brian Major,¹
Brian McLean,³ Oliver Oberdorf,³ and David R. Rodriguez³

¹*Canadian Astronomy Data Centre, National Research Council Canada,
Victoria, British Columbia, Canada*

²*ESAC Science Data Centre, Villanueva de la Cañada, Madrid, Spain*

³*Space Telescope Science Institute, Baltimore, Maryland, United States of
America*

Abstract. The Canadian Astronomy Data Centre (CADC) and the European Space Astronomy Centre (ESAC) maintain mirrors of and provide user access to the HST data collection. A new mirroring approach was needed to improve consistency and support future missions like JWST. The Common Archive Observation Model (CAOM) is used as the core model for all data holdings at the CADC and the Mikulski Archive for Space Telescopes (MAST) and was extended to support a metadata and data synchronization system.

The metadata synchronization process relies on a simple RESTful web service operated by the metadata source (MAST) and a metadata harvesting tool run by the mirror centers (CADC and ESAC). The data synchronization process uses CAOM metadata to discover and retrieve files from the source (MAST) to the mirror sites. Through the use of a backend plugin, the CADC and ESAC have extended the file synchronization tool to interface with their respective site-specific storage systems.

Using the common metadata model, services and tools as a base, partners can augment their own system with additional information specifically intended to provide added value features. ESAC provides information about publications in their instance of the Archive and both CADC and ESAC provide additional IVOA services for users.

1. CAOM Enhancements

This project required several enhancements to CAOM and resulted in the release of CAOM version 2.3 in 2018.

Several changes were designed to improve extensibility so that metadata curators could better describe their content and use CAOM as their primary data model.

We lifted a constraint on the algorithm name for a SimpleObservation so that other values (e.g. simulation) could be used and retained exposure as the default value.

CalibrationLevel was modified to support new values in IVOA ObsCore (Louys et al. 2011) (calib_level) and extended to include planned observations. The DataProductType was converted from an enumeration to an extensible vocabulary with the base vocabulary terms taken from ObsCore (dataprodukt_type); as a vocabulary, new terms

can be introduced simply by providing a resolvable URI (typically an http URL) referring to the definition of the new term(s). The ProductType enumeration was also converted to a vocabulary with the base terms from the CAOM-2.2 enumeration values.

Polygon validity (used to describe the spatial coverage of an observation) was restricted to be consistent with IVOA DALI (Dowler et al. 2017) polygon (simple polygon with counterclockwise winding direction). The previous polymorphic shape and general polygon with disjoint parts was retained to provide a more detailed footprint when necessary.

A content checksum was added to the Artifact class to support data Synchronization. In CAOM checksums are expressed as URIs where the scheme is the checksum algorithm and the scheme-specific part is an ASCII representation (hexadecimal) of the value. We are currently using MD5 checksums for file verification as it provides a good balance of robustness and performance and was supported already by the CADC archive storage system.

In order to support robust metadata synchronization and validation, we defined a metadata checksum algorithm so that each serialized entity included a stable checksum of all metadata fields and an accumulated metadata checksum of itself and all child entities. The accumulated metadata checksum of the Observation thus captures the state of the entire structure and can be used to verify serialization, transmission, and deserialization of a complete observation. This checksum protects against a variety of software bugs: lossy storage of numeric values in databases and files, process flaws like updating database values without updating corresponding checksums, inconsistent reading and writing of XML documents, inconsistent handling of lists or ordering of sets, and was even used to detect and fix a race condition effecting database updates.

2. Metadata Synchronization

Metadata synchronization involves an application (caom2harvester) and a web service that implements a simple REST (Fielding 2000) API. The web service API supports an endpoint with path elements for the collection and the observationID from CAOM (/observations/collection/observationID). A GET request to the collection returns a list of observations; optional parameters allow control of start and end timestamp values and to limit the number of records in the list. The listing includes the collection, observationID, modification timestamp, and accumulated metadata checksum of the observation in order of increasing modification timestamp. A GET request to the complete path returns a CAOM observation document (XML). There is a similar endpoint that provides a list of deleted observations (/deleted/collection); the output format is similar to the observation listing, but it includes the internal UUID of the observation and does not include a metadata checksum.

The caom2harvester tool harvests a single collection at a time and normally operates in incremental mode (recent changes) to maintain an up-to-date copy of the metadata. The read-only web service is implemented at MAST while CADC and ESAC operate the caom2harvester application to pull metadata updates from MAST and store in a local database (currently PostgreSQL). For the project, CADC and ESAC could harvest metadata from MAST at approximately 50K observations per hour (significantly faster than the MAST pipeline could generate them) so keeping up to date was easily accomplished with a single cron job.

The caom2harvester keeps track of current harvest state (last timestamp seen) and all failures in the destination database, so it can be killed or crash and be restarted with no difficulty. A retry mode can be used to reharvesting of tracked failures. The outcome of a retry will be one of fix, delete, or fail again.

The caom2harvester tool also supports a validation mode to check and maintain the consistency of the entire metadata collection. This mode gets a complete listing from the source (REST API) and local (database) and makes three comparisons:

- missing observations (in source - not in destination)
- missed deletions (not in source - in destination)
- accMetaChecksum mismatch (cause: persistence or serialization bug in source or destination or harvest is not up to date)

In practice, incidents detected by validation are used to create new failure records and then the retry mode is used to reharvest those and bring the destination database up to the correct state.

3. Data Synchronization

For this project, we implemented a new file synchronization tool named caom2-artifact-sync that used the local CAOM database to discover files to retrieve from the source. The discovery mode of this tool uses the same observation timestamps to scan new or changed observations and figure out which artifacts (files) to retrieve. To do this, the tool makes use of a local storage plugin (plugin API defined) to check the current file status (existence and checksum).

Like the metadata harvesting tool, file synchronization normally operates in incremental mode: it uses local CAOM metadata to discover new or modified files and schedule downloads. A separate mode performs downloads. A validation mode performs a full comparison of files referenced in CAOM with those in the local storage system and (optionally) schedules downloads to fix any discrepancies. Since CADC and ESAC are only able to service public data, downloads are initially scheduled for the data release date found in the CAOM metadata. Failed downloads are re-scheduled with a delay that depends on the type of error encountered). The download mode is multi-threaded and we found that 16-48 threads were needed to attain good network utilization, largely due to overheads in retrieving small files.

After some network tuning at MAST, CADC and ESAC were able to retrieve 30TiB per day for the initial downloads; incremental updates due to reprocessing at MAST are picked up and synchronized with minimal effort.

The caom2-artifact-sync tool also supports validation mode. This mode compares the local CAOM artifact list to storage content to detect missing files or files with checksum mismatch (reschedule download) and to detect orphaned files in storage (due to a rename at MAST or artifact being removed from CAOM) and schedule deletion.

4. Data Delivery to Users

CADC and ESAC provide HTTP download service that uses local CAOM database to determine: if the file exists, if the file is readable by the user (public in the case of HST),

and if the local copy is up-to-date (`Artifact.contentChecksum == local storage checksum`): deliver the file. Otherwise, if the file exists we redirect the download request to the MAST download service (same as used in data sync above)

Future work: All partner sites could use client location (geo-ip) to redirect downloads to another partner, thus implementing a sort of content distribution network (CDN) for shared and synchronized data collections.

5. Site-Specific Enhancements

For operational support, CADC makes the table of metadata harvest failures visible via a TAP (Dowler et al. 2010) service; MAST can diagnose and fix metadata issues and once incremental harvesting picks up the changes the failures are automatically removed. CADC also makes the table of scheduled downloads (including data sync failures) visible via the same TAP service; MAST can diagnose and fix data delivery issues and eventually see the results.

For users, MAST makes publication metadata available and ESAC harvests this and combines it with the CAOM metadata in the EHST interface.

The MAST portal uses the CAOM database to search the HST metadata. The CADC AdvancedSearch and the ESAC EHST portals provide alternate browser-based user interfaces to search the HST metadata.

CADC provides IVOA services TAP, SIA (Dowler et al. 2015b), DataLink (Dowler et al. 2015a), SODA (Bonnarel et al. 2016), and IVOA ObsCore data model support as part of their programmatic archive data discovery services. ESAC also provides IVOA services TAP, SIA, SSA (Tody et al. 2012) based on the HST metadata in their CAOM database.

References

- Bonnarel, F., Demleitner, M., Dowler, P., , Tody, D., & Dempsey, J. 2016, IVOA server-side operations for data access 1.0, IVOA Proposed Recommendation 20 September 2016. URL <http://www.ivoa.net/documents/SODA/>
- Dowler, P., Bonnarel, F., Michel, L., & Demleitner, M. 2015a, IVOA datalink 1.0, IVOA Recommendation 17 June 2015. URL <http://www.ivoa.net/documents/DataLink/>
- Dowler, P., Demleitner, M., Taylor, M., & Tody, D. 2017, Data access layer interface, version 1.1, IVOA Recommendation. URL <http://www.ivoa.net/documents/DALI>
- Dowler, P., Rixon, G., & Tody, D. 2010, Table access protocol version 1.0, IVOA Recommendation. URL <http://www.ivoa.net/documents/TAP>
- Dowler, P., Tody, D., & Bonnarel, F. 2015b, Ivoa simple image access, version 2.0, IVOA Recommendation 23 December 2015
- Fielding, R. T. 2000, Doctoral dissertation, University of California, Irvine. URL <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- Louys, M., Bonnarel, F., Schade, D., Dowler, P., Micol, A., Durand, D., Tody, D., Michel, L., Salgado, J., Chilingarian, I., Rino, B., de Dios Santander, J., & Skoda, P. 2011, Observation data model core components and its implementation in the Table Access Protocol, version 1.0, IVOA Recommendation. URL <http://www.ivoa.net/documents/ObsCore/20111028/REC-ObsCore-v1.0-20111028.pdf>
- Tody, D., Dolensky, M., McDowell, J., Bonnarel, F., Budavari, T., Busko, I., Micol, A., Osuna, P., Salgado, J., Skoda, P., Thompson, R., & Valdes, F. 2012, Simple spectral access protocol version 1.1, IVOA Recommendation. URL <http://www.ivoa.net/documents/SSA/20120210/REC-SSA-1.1-20120210.htm>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Mapping Data Models to VOTable

Omar Laurino,¹ Gerard Lemson,² Mark Cresitello-Dittmar,¹ Tom Donaldson,³
and Laurent Michel⁴

¹*Smithsonian Astrophysical Observatory, Cambridge, MA, USA;*
olaurino@cfa.harvard.edu

²*Johns Hopkins University, Baltimore, MD, USA*

³*Space Telescope Science Institute, Baltimore, MD, USA*

⁴*Observatoire de Strasbourg, Strasbourg, France*

Abstract. Data providers and curators provide a great deal of metadata with their data files. This metadata is invaluable for users and for Virtual Observatory software developers. In order to be interoperable, the metadata must refer to common data models. A new specification is being developed by the IVOA Data Modeling Working Group to define a scheme for annotating VOTable instances in a standard, consistent, interoperable fashion, so that each piece of metadata can unambiguously refer to the correct data model element it expresses. The mapping can be extended to formats other than VOTable.

1. Use Cases

The use cases for a serialization format may be rather abstract and technical, because the format is generally agnostic of the data that it will encode. However, a number of domain specific constraints can be taken into account to drive the format's design. Similarly, one can spell out exemplary domain use cases that make it explicit what kind of usage the serialization format addresses.

In the most abstract sense the main use case is to provide a complete, unambiguous, interoperable means to encode and decode instances of structured data models in VOTable and other existing formats. This is something that the current formats do not allow, or at least not in an interoperable and unambiguous fashion, or they are not expressive enough to represent complex structured instances (Thomas et al. 2015; Graham et al. 2013). Moreover, the format must be extensible, in the sense that users must be able to express instances of new models with a single standard, rather than having to come up with a standard representation for each model.

Note that the above are rather strong constraints, because they entail that the format requires a *mapping* between two existing representations. One representation defines the model and its instances, the other is the existing data format, e.g. VOTable.

The constraints above lead to an unusual data modeling workflow. Typically an abstract representation of a model is then translated to a physical, specific representation through object oriented software, data base schemata, or XML schemata. In our case we are looking for a way to *annotate* data and metadata in existing formats with their

own pre-existing schemata, so that such data and metadata can be interpreted in terms of complex concepts in a generic, interoperable, shared conceptual data model.

A typical usage scenario may be a VOTable client that is sensitive to certain models only, say coordinates, measurements, and World Coordinate System transforms. Such a client may be written to understand annotations for coordinates and frames, manipulate such instances, and write them back to disk. The actual representation is usually tabular, with e.g. coordinates expressed as cells in a table whose columns represent several properties of a source, including its position.

More complex models for astronomical data products may enable smart plotting and fitting applications. As high-level models for Data Cubes, Time Series, or Spectra use the same building blocks for measurements and coordinates an application may discover these pieces of information and structure a plot, or perform a fit, or create an animation, with minimal user input, together with features for data validation, units conversion, intelligent projections or domain specific calculations.

Having a generic standard for annotating VOTable and other tabular formats can be useful even when clients are not necessarily aware of all or even any of the models mapped in a data file. For instance, data discovery portals may provide a friendly interface that allows users to select or query by physical quantities using standardized, structured representations. It may do so dynamically for all the pieces of metadata present in the dataset, rather than limiting functionality to a set of hard-coded metadata properties.

Moreover, the existence of an explicit data model representation language and of a precise mapping specification enables the creation of universal validators, just as it happens for XML and XSD: the validator may parse the data model descriptions declared by the VOTable and check that the file represents valid instances of one or more data models.

While most use cases enabled by this specification may help clients implementing useful features for users, the definition of a standard for serializing data model instances can lead to the development of tools for publishers to build templates of valid responses. The mapping specification thus provides data publishers with a framework for expressively and accurately representing their data holdings in an interoperable way, thus leveraging all the work that goes into maintaining and curating their data and metadata.

2. Mapping Data Models to VOTable

Designing an interoperable scheme for representing structured objects in multiple contexts interoperably leads to the challenges usually associated with Object-Relational Mapping.

The VO Data Modeling Language (Lemson et al. 2018) defines a very simple XML schema for representing data models as XML documents¹. VODML maps directly to a very small subset of UML, so that each data model can be represented by a simple UML class diagram. Instances of data model types could thus be represented by UML object diagrams, although that is not usually useful and certainly not part of the standard. In practice such instances are usually represented in relational databases,

¹Note how this is different from the usual workflow where data models *are* indeed represented by XML schemata, rather than document

object-oriented software, FITS, VOTable, and other tabular or non-tabular representations.

The inherent complexity of Object-Relational mapping is mitigated by the fact that in most cases astronomical data sets are presented to the user in a rather denormalized, flattened fashion as a single table. This allows for a simple annotation for simple cases, which are the vast majority.

Complex representations with multiple tables must still be supported for more complex cases, especially for those where instances in a table may refer to instances in other tables. While this is not a common scenario, there are indeed complex data products where a single table can, for instance, contain multiple time series.

VODML Data Models define *Types* and *Roles*, e.g. `EquatorialCoordinate` is a *Type* with *Roles* `ra`, `dec`, and `frame`. Models define globally unique, portable identifiers for types and roles, e.g.² `coords:domain.space.EquatorialCoord` and `coords:domain.space.EquatorialCoord.frame`.

These portable identifiers can be used to map data and metadata in VOTable to the concept they represent, in an unambiguous fashion. The full specification of how to perform such mapping is the subject of the Mapping Models to VOTable standard (Lemson et al. 2017).

The mapping specification defines an extension to the VOTable v1.3 schema that provides hooks for data producers to annotate their tabular data files.³

The annotation for the full file is all contained in a single *VODML* block, so it's easy for non-VODML or legacy clients to simply ignore the new annotations. Additionally, data producers can add annotations without changing their existing files or services. Indeed, one could even add the *VODML* element with its annotations programmatically in order to wrap existing data files, given that a mapping is performed once for each class of data files. Enabling such mapping of existing services or files is part of the reference implementations described in the next section.

At the time of this writing VODML is an actual IVOA standard, while the Mapping standard is currently in its Working Draft stage, as implementations inform changes and design decisions to the first drafts of the document.

3. Implementations

A web page (<https://olaurino.gitlab.io/ivoa-dm-examples>) collects information and demonstrations of many such implementations⁴, including data models, software, and notebooks demonstrating the interoperability of independent reference implementations on the server and client side. The serializations include coordinates, measurements, and time series as sparse data cubes.

²These examples are based on current models but they are not necessarily actual identifiers from actual models. They are just realistic examples.

³An equivalent mapping is possible with the basic VOTable 1.3 schema only. However, after an initial round of implementations the working group decided to review the VOTable schema to make the annotations simpler and more flexible.

⁴Given the ongoing nature of these implementations, the best pointers we can provide at this point are URLs

A service providing VODML-annotated files was developed at the Space Telescope Science Institute as part of a development version of the MAST portal, while a Python client (<https://github.com/olaurino/rama>) implementing the Mapping specification was developed at the Chandra X-Ray Center.

The Python client (Rama) offers a framework for translating VO standard representations as Python objects, with a specific focus on leveraging Astropy (Greenfield et al. 2013) classes and features.

Creating valid annotations can be an error prone process. However, the existence of consistent standards for both model descriptions and annotations makes it possible to write generic tools that abstract users from standard documents, so to allow them to create annotations using their domain knowledge and the knowledge of their data sets. Two tools have been implemented with very different philosophies, as described below.

Jovial (<https://github.com/olaurino/jovial>) implements both VODML and the Mapping standard to provide data modelers and providers with Domain Specific Languages (DSLs). The first implementation was written in Groovy, while a new implementation in Kotlin, with better support for Integrated Development Environments, is underway. Domain Specific Languages can be useful in reducing the complexity of tasks by employing a more human readable and writable syntax for writing programs in a more natural, declarative way.

While DSLs are rather useful because a textual format is easily versionable, shareable, and allows for rapid prototyping, Graphical User Interfaces (GUIs) offer the shortest learning curve possible. For this reason a *drag and drop* GUI was developed for allowing users to easily map standard data models to tables and even services. The application can help users and data providers to create template or even full annotations for their datasets and services while requiring minimal exposure to the standard document themselves.

Acknowledgments. Omar Laurino and Mark Cresitello-Dittmar acknowledge support by NASA under contract NAS 8-03060 to the Smithsonian Astrophysical Observatory for operation of the Chandra X-ray Center. Gerard Lemson is funded by the U.S. National Science Foundation through its Data Infrastructure Building Blocks (DIBBs) program, award ACI-1261715. Tom Donaldson acknowledges support from the NASA Astronomical Virtual Observatories (NAVO), provided by NASA through the Astrophysics Data Curation and Archival Research (ADCAR) program. Laurent Michel is funded by OV France and by the the Survey Science Consortium of XMM Newton.

References

- Graham, M., et al. 2013, UTypes: current usages and practices in the IVOA, Tech. rep.
Greenfield, P., et al. 2013, Astropy: Community Python library for astronomy, Astrophysics Source Code Library. 1304.002
Lemson, G., et al. 2017, Mapping Data Models to VOTable, Tech. rep.
— 2018, VO-DML: a consistent modeling language for IVOA data models, Tech. rep.
Thomas, B., et al. 2015, Astronomy and Computing, 12, 133

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

The New Science Portal and the Programmatic Interfaces of the ESO Science Archive

A. Micol,¹ M. Arnaboldi,¹ N. Delmotte,¹ V. Forchì,¹ N. Fourniol,¹
O. Hainaut,¹ U. Lange,¹ A.M. Kahn,¹ L. Mascetti,² J. Retzlaff,¹
M. Romaniello,¹ D. Sisodia,³ C. Spiniello,¹ M. Stellert,⁴ F. Stohr,¹ I. Vera,¹
and S. Zampieri¹

¹*European Southern Observatory, Garching bei Muenchen, Germany;
amicol@eso.org*

²*Terma GmbH, Germany*

³*Pactum GmbH, Germany*

⁴*Tekom GmbH, Germany*

Abstract. In June 2018 the new ESO science archive interfaces have become available to the astronomical community. Powerful new features allow a much richer user experience than ever before. The two main components are the ESO Archive Science Portal for interactive web access, and the VO-based programmatic and tool access.

The Science Portal provides a web-based interface to browse and explore the archive with interactive, iterative queries. The results are presented in real time in various tabular and/or graphic forms, including interactive previews, allowing an evaluation of the usefulness of the data which can then be selected for retrieval.

The direct database and Virtual Observatory layer allows a more flexible and customizable access allowing users to perform complex queries, to script their access, or to use common VO-aware tools to access the wealth of data in the ESO Science Archive. Extensive documentation is provided in terms of practical examples, which are intended to provide templates for users to customise and adapt to their specific needs.

In this first release, the Science Portal supports processed data from the La Silla Paranal Observatory, while the programmatic layer already supports processed, raw, and ambient data. Future plans include: support for ALMA processed data by both components, and Science Portal support for raw data. It is planned that these new access points will gradually replace the previous ones for La Silla Paranal data, while ALMA will keep maintaining a dedicated, separate access.

1. The ESO Archive Science Portal

The most immediate way to access the new archive services is through a web application, the ESO Archive Science Portal¹. It is an Angular application that uses the RESTful Elasticsearch search engine, based on the Apache Lucene library. An in-house developed plugin (Astroes, Vera, I. 2017, ADASS XXVII) is used to provide the

¹<https://archive.eso.org/scienceportal/>

necessary spatial query capabilities on the unit sphere. The Science Portal presents to the users three main views:

- the sky view (based on Aladin Lite, CDS)
- the tabular view
- the aggregation view

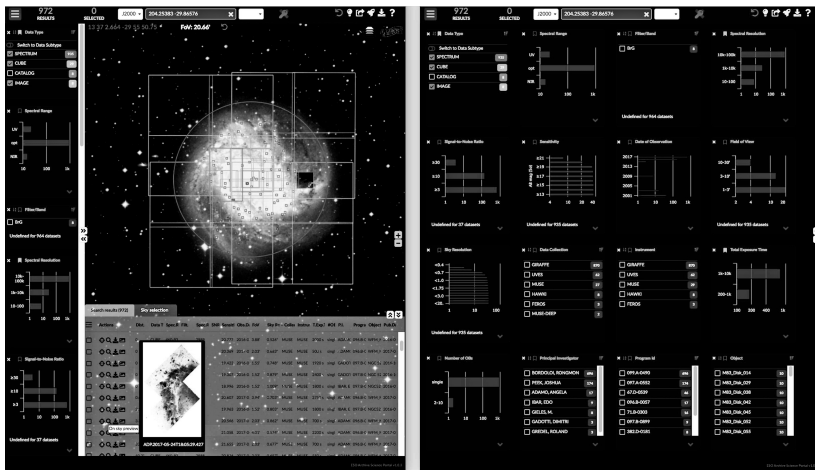


Figure 1. The ESO Archive Science Portal (left), and the fully opened aggregation view (right)

Within the *sky view*, the footprints of the query-matching datasets are graphically displayed over a full-sky background imagery (default: DSS2 coloured). In case the number of matching datasets exceeds 1000, a spatial density map of the matching results replaces the drawn footprints to avoid too crowded a representation. Footprints are available for images, cubes, catalog tiles, and for some of the catalogs; spectra and visibilities are represented instead by a point in the sky.

Within the *aggregation view*, the possible values that a query parameter can take are grouped and presented as histograms or lists, as appropriate. In this way, the system communicates its content to users at all times, without the need for any previous knowledge. The query constraints can be specified here by explicitly entering them and/or by selecting values or ranges arranged in lists (for string-valued fields) or histograms (for numeric fields). Also, to remove the need of *a priori* knowledge, for string-valued parameters (e.g. name of the principal investigator (PI)), auto-completion is provided.

Within the *tabular view*, the main characteristics of the matching datasets are displayed in text format. Within this view the user has control over other features, like: getting to display the preview of a dataset, either in a pop-up window (activated by a mouse-hover), or loaded within the sky view itself (e.g. for images); getting full details of a particular dataset, including an interactive preview also for spectral data, etc.

Each of the views can be hidden or expanded to full screen, for improved ergonomics.

1.1. Multi-dimensional faceted search

In order to serve a broad range of use cases, 18 query parameters are available. They are a combination of positional parameters (cone search around a given position on the sky), physical characteristics of the data (for example, signal-to-noise ratio, sensitivity, spectral range, spectral and spatial resolution), the observational setup (for example, filter name and exposure time), the ESO observing process (for example, Principal Investigator name and Programme ID), and the data type (one of: image, cube, spectrum, source table, catalog, visibility).

A facet shows to the user how often a chosen enumerated list of the parameter's values are present in the result set (e.g., the Data Type facets shows that there are 2 million spectra, 400,000 images, 7000 cubes, etc.). Faceted navigation provides the guidance missing in parametric searches, allowing users to elaborate queries progressively; it allows users to see the impact of each incremental choice in one facet on choices in other facets. In fact, posing a constraint on any parameter makes the interface recomputing all the facets, so that the user can have always the up-to-date view of the distributions of all the parameters for the matching datasets.

2. The direct database and Virtual Observatory access

The user who wants to perform complex queries, or access the archive via scripts, or via existing tools, can do so through the new programmatic and tool access layer.

This layer provides direct database access via the Astronomical Query Data Language (ADQL), and the Tabular Access Protocol (TAP) of the International Virtual Observatory Alliance (IVOA). Also provided are the Simple Spectral Access protocol (SSA), and the so-called ObsCore standard service.

Among the VO standards, *ObsCore* provides the highest level of interoperability with respect to science data discovery: an astronomer can prepare a standard query to find data of interest, and send it to TAP services at multiple sites, to perform global data discovery without having to understand the details of the services present at each site.

Finally, the *DataLink* standard service allows the user to find all files related to a given dataset, like provenance, ancillary files, previews, data release descriptions, etc. The relevant DataLink link is usually provided in the response of a TAP query.

2.1. The VO choice and the infrastructure

The choice of implementing the programmatic layer based on VO standards is based on two main reasons: (1) the very pragmatic opportunity to re-use existing software libraries and approved standards, greatly reducing both the design and the development phases, and (2) the fact that standard protocols allow for interoperability, augmenting the discoverability of the ESO science data.

The end points of the VO protocols (TAP, DataLink, SSA) have been registered at the <http://registry.euro-vo.org/> under the ivo://eso.org/ authority². This allows discovery agents, registry-aware tools, as well as software developers, to easily find and connect to the new programmatic layer of the ESO Science Archive services.

²All VO identifiers are of the form: ivo://eso.org/*.

The programmatic layer is based on two different database management systems: MS SQL Server for full support of the spatial queries on the observed data (tap_obs service), and SYBASE IQ for accessing large astronomical catalogues (the largest to date has got more than 31 billion records), though with limited spatial query support (tap_cat service).

2.2. What's there for the end-user

The above-described back-end provides to the end-user astronomer a number of powerful new functionalities:

- Support for complex queries:
 - spatial queries on the footprints of the processed data (*e.g.*, queries to find data whose footprint contains or intersects a specific region, or another footprint)
 - joins and sub-queries on different tables (*e.g.*, linking processed data with the ambient measurements at the time of the raw observations)
 - queries with sequences of logical operators (OR, NOT, AND)
- Access the science archive directly via VO-aware tools, *e.g.*, TOPCAT, Aladin, SPLAT-VO.
- Scripting repetitive tasks (*e.g.* in a cronjob) to query and download data from the ESO Science Archive can be easily achieved using externally-developed common software libraries and packages, *e.g.* pyvo (Python), astroquery.utils.tap (Python), stilts (Java).
- Direct file download is now allowed on public files, without the need to submit and handle an archive request.
- A tutorial page exists³ with modifiable and runnable examples of common ADQL queries, with a growing list of Python scripts, with examples on how to manage your asynchronous TAP requests, etc. Both users and developers are invited to have a look at it to learn how to interface programmatically to the ESO Science Archive.

3. ESO Archive Community Forum

The *ESO Archive Community Forum* <https://esocommunity.userecho.com/> is a platform for sharing ideas and methods, asking questions and sending feedback and suggestions on how to improve and use the new ESO Archive Science Portal and on how to gain programmatic and tool access to the archive science portal. Contributions are welcome from the users, also without any need for registration, they are pre-approved, monitored and moderated by the ESO Archive Science Group (ASG).

Acknowledgments. We would like to thank and provide credits to: Aladin Lite (CDS, Strasbourg), TAPLIB (G. Mantelet, ARI/CDS), taplint (M. Taylor, Physics, Bristol University, UK), pyvo (ARI, Heidelberg).

³<https://archive.eso.org/programmatic/>

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Science Exploitation in a Big Data Archive: the Euclid Scientific Archive System

Sara Nieto,¹ Pilar de Teodoro,¹ Fabrizio Giordano,¹ Elena Racero,¹ Monica Fernandez,¹ Damien Noiret,² Jesus Salgado,¹ Bruno Altieri,¹ Bruno Merin,¹ and Christophe Arviset¹

¹*European Space Astronomy Center, European Space Agency,
Spainsnieto@sciops.esa.int*

²*ESILV Leonardo da Vinci Engineering School, Courbevoie, France*

Abstract. Euclid is an ESA mission and a milestone in the understanding of the geometry of the Universe. The Euclid Archive System (EAS) is a joint development between ESA and the Euclid Consortium and is led by the Science Data Centres (SDC) of the Netherlands and the ESDC (ESAC Science Data Centre). Big-data technologies, driven by data nature and volume, are transforming the way of doing scientific research towards collaborative platforms whose first goal is to enable access and process large data sets in ways that could not be done downloading the data. Some examples of the main technologies explored as part of the Euclid scientific archive are: JupyterLab, Apache Spark, GreenPlum and PostgresXL.

1. Introduction

The Euclid mission (Laureijs et al. 2011) will map the sky in a single optical band and three near-infrared bands (H, J and Y). It will measure photometric and spectroscopic redshift of galaxies to understand the properties and nature of dark matter and dark energy. Euclid will be launched in 2022 and will complete a wide survey (covering 15000 deg^2) and a deep survey (covering 40 deg^2 and 2 magnitudes deeper than the wide survey) during 5 and a half years of observations. During the nominal operations phase, Euclid will combine the space survey with ground-based surveys to achieve its scientific objectives. This will boost the data volume produced by Euclid SGS up to 26PB per year and a catalogue up to 10 billion objects (Pasian et al. 2014).

In terms of organization, the Euclid Science Ground Segment (SGS) is a distributed data processing and data storage system, which is responsible for processing the data from ingested raw frames to science-ready images, spectra and catalogs and deliver them to ESA (Pasian et al. 2014). According to requirements on the SGS the design of the Euclid Archive System was established as a combination of 3 independent subsystems: the Data Processing System (DPS) consists of metadata storage and services which support the data processing inside the SGS; the Science Archive System (SAS) which is a gateway for end-users to Euclid data and supports the scientific use-cases, the release delivery of data to the wide astronomical community, and long-term data preservation; the Distributed Storage System (DSS) consists of data files storage for both the DPS and SAS.

2. Scientific Archive System

The SAS aims to support the scientific exploitation of the most valuable Euclid data for the Euclid Consortium (EC) and the wider astronomical community. The SAS is currently under development at the ESAC Science Data Centre (ESDC), which is responsible for the development and maintenance of the scientific archives for the Astronomy, Planetary and Heliophysics missions of ESA. The design and architecture of the SAS follows the latest technology generation of archives developed by the ESDC (Martinez, B. and others 2018), taking full advantage of the existing knowledge, expertise and software libraries from the Gaia Archive (Gonzalez, J. and others 2018) among others.

The functionalities of SAS are mainly focused on exploitation of catalogues and spectra as well as visualization of Level 2 maps. For that purpose, the archive will provide a Graphical User Interface (GUI) as guided access for parametric search on metadata and catalogues together with a visual interface for maps exploitation based on ESASky technology (Racero, E. and others 2018). In addition to the GUI, the SAS will also provide with a set of Application Programming Interfaces (API) Virtual Observatory compliant to access the data programmatically.

Given the huge volume of data that will be generated, in the order of petabytes, and the heterogeneous nature of scientific products that the mission will produce (e.g. images, catalogues and spectra), we started a technology exploratory phase mainly focused on enabling scientific use cases provided by the Euclid Archive Users Group (EAUG) composed by EC and ESA members.

Among the science driven use cases studied, we are focused on the analysis of large catalogues and pixel data volumes. We based the study on the analysis of 10 billion sources catalogues and hundreds of columns on Massive Parallel Processing (MPP) databases like PostgresXL (<https://www.postgres-xl.org/>) and GreenPlum (<https://greenplum.org/>), while Apache Spark (<https://spark.apache.org/>) is intended to be used for the processing of petabytes of pixel data.

3. MPP Databases

There will be a need for scaling out our database system as the amount of data to be stored for querying will be very large. Different options for distributing PostgreSQL have been evaluated: Postgres-XL, CitusData and GreenPlum. The most solid one currently is Greenplum, which is a massive parallel data analysis system based on PostgreSQL. Greenplum allows to scale out easily and query the data through the Table Access Protocol (TAP) (Dowler et al. 2011). The tests performed so far takes into account the performance, maintainability and scalability.

The tests ran were performed in a private cloud environment. For Postgres-XL we used 10 datanodes, for CitusData 3 nodes and for commercial Greenplum 6 datanodes with 4 segments on each. All with 32GB of RAM and shared storage (NetApp <https://www.netapp.com/>) for storing the data.

We found some issues that stopped us continue testing on Postgres-XL as the data reshuffling was not possible using 1TB tables and a JDBC bug not resolved yet that prevented us using it on the SAS. CitusData performance was much worse as it was not using the advantages of postgres 10 at that moment, so we discarded it as well. Greenplum, on the other hand, performed correctly the expansion to new nodes proving that the scalability with big tables was real.

The performance achieved was better than in a single instance of postgres for big tables, for small table the queries were resolved on memory, but it was observed that the distribution key must be carefully selected to achieve the best performance. Making the distribution key a healpix index will allow us to partition the data in a way that will be optimal to perform crossmatches.

High availability tests were done with greenplum showing good recovering with little interaction. The recommended hardware for MPP databases is to run on commodity hardware and local disks, preferably SSD. Those tests are envisaged for the near future to evaluate the hardware needed for the project for this kind of databases.

4. Apache Spark

The main objective of the study was to clarify the feasibility of using this framework as a backend for the analysis of astronomical data and fulfil the main use cases identified as the drivers of the proof of concept: interactive analysis of large catalogues and pixel data.

Apache Spark is a distributed parallel processing framework widely used in the industry for analysis of large datasets, it is open-source and it is based on the MapReduce strategy popularized by Hadoop. Spark is a cluster computing solution which provides a stack of libraries to work with structure data (SparkSQL), streaming applications (SparkStreaming) and machine learning algorithms (MLib) among others. In addition, Spark provides APIs for Python, Java and R, Python being the language most widely used within the astronomical community.

In the first phase of the study we defined a preliminar cluster architecture of 6-workers of 32GB RAM and 8 cores each in standalone mode and NetApp (NFS) virtual volumes as storage solution. As target dataset we used a simulated catalogue of 2.7 billion sources with a hundred columns that was migrated from CSV to Parquet, which is the default format in Spark v2.x. Apart from Parquet, it accepts other formats like Comma Separated Values (CSV), JavaScript Object Notation (JSON) and other external data sources. For the migration process and given the huge dataset, we took into consideration a set of parquet parameters in order to optimize the usage of the storage volume and balance between the compression overhead and I/O.

Once the catalogue was migrated, we perform a set of SparkSQL-compatible queries like coordinates box searches, parametric selection, count operations and rows sorting to evaluate the users workflow and performance metrics. The results showed that filter pushdown mechanism improves query performance by reducing the data loaded, with the storage being the bottleneck.

As part of these tests, we evaluated how to work with catalogues and images in FITS format for which we used Spark-FITS connector (Peloton et al. 2018) implemented by AstroLab as Spark does not support FITS format natively. Spark-FITS connector allowed us to analyze FITS formatted catalogues and source extraction on simulated images in a seamless way for final users.

Finally, the results of these preliminary tests show that Spark can be potentially used to perform interactive analysis on big astronomical catalogues, but at the same time we detect that there is still work to do around Spark ecosystem of tools and libraries in order to make wide use of Spark for Astronomy.

5. Future work and Conclusions

In the next phase we will continue performing the evaluation addressing the tests on bare metal for performance analysis. From the users point of view, interactive analysis on Jupyter Notebooks is becoming the trend in the astronomical community, therefore the assessment of these technologies connected to the data archive is already in our roadmap. In addition and as a future step, the evaluation of GreenPlum and Apache Spark opens the door to the execution of Machine Learning algorithms directly on the data coming from the Euclid mission enabling high valued applications like automatic classification.

The analysis of big datasets requires state-of-the-art technologies in order to fulfill the scientific astronomical use cases demanded by the community. In this sense, we are immersed in a proof of concept phase to adopt the most suitable technologies to enable data discoveries on Euclid cosmological objectives: weak lensing and galaxy clustering. In the core of this study phase GreenPlum on database side and Spark as analysis platform are currently under evaluation.

References

- Dowler, P., Rixon, G., & Tody, D. 2011, arXiv. [arXiv1110.0497](#)
- Gonzalez, J. and others 2018, in ADASS XXIX, edited by N. P. F. Lorente, & K. Shortridge (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD
- Laureijs, R., et al. 2011, arXiv. [arxiv1110.3193](#)
- Martinez, B. and others 2018, in ADASS XXIX, edited by N. P. F. Lorente, & K. Shortridge (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD
- Pasian, F., Hoar, J., Buenadicha, G., Dabin, C., Sauvage, M., Poncet, M., Noddle, K., Delouis, J., & Mansutti, O. 2014, in Astronomical Data Analysis Software and Systems XXIII, edited by N. Manset, & P. Forshay, vol. 485 of Astronomical Society of the Pacific Conference Series, 505
- Peloton, J., Arnault, C., & Plaszczynski, S. 2018, arXiv. [arXiv1804.07501v2](#)
- Racero, E. and others 2018, in ADASS XXIX, edited by N. P. F. Lorente, & K. Shortridge (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

The VLITE Database Pipeline

E. Polisensky,¹ E.E. Richards,² T. Clarke,¹ W. Peters,¹ and N.E. Kassim¹

¹*Naval Research Laboratory, Washington, DC, USA;*

Emil.Polisensky@nrl.navy.mil

²*ATA, LLC, Vienna, VA, USA*

Abstract. A post-processing pipeline to adaptively extract and catalog astronomical sources has been developed to enhance the scientific value and accessibility of data products generated by the VLA Low-band Ionosphere and Transient Experiment (VLITE) on the Karl G. Jansky Very Large Array (VLA). In contrast to other radio sky surveys, the commensal observing mode of VLITE results in varying depths, sensitivities, and spatial resolutions across the sky based on the configuration of the VLA, location on the sky, and time on source specified by the primary observer for their independent science objectives. Previously developed tools and methods for generating source catalogs and survey statistics proved inadequate for VLITE's diverse and growing set of data. A raw catalog of sources extracted from VLITE images is created from source fit parameters stored in a queryable database. Sources in the raw catalog are associated with previous VLITE detections in a resolution- and sensitivity-dependent manner, and cross-matched to other radio sky surveys to aid in the detection of transient and variable sources. Final data products include separate, tiered source catalogs grouped by sensitivity limit and spatial resolution.

1. The VLA Low-band Ionosphere and Transient Experiment: VLITE

In late 2014 the Remote Sensing Division at the US Naval Research Laboratory secured funding for a commensal experiment to operate using the newly re-designed low frequency system at the VLA. The scientific motivations include astrophysical imaging, (Clarke et al. 2016) serendipitous transient radio astronomy (Polisensky et al. 2016) and ionospheric remote sensing (Helmholtz et al. 2015). VLITE¹ was conceived as a prototype system with dedicated samplers and fibers that tap the signal from a subset of VLA P-band receivers and correlate them through a dedicated software correlator. The usable bandwidth is restricted to an RFI-free 40 MHz centered at 340 MHz. Providing over 6000 observing hours per year, VLITE began full science operations in November 2014 on 10 antennas and expanded to 16 antennas in the summer of 2017. VLITE operates nearly continuously and independent of the VLA on-line system except that its pointings are slaved to the Cassegrain science program at high frequencies 1 – 50 GHz.

At the end of each UTC calendar day the data are sorted into datasets based on the primary observing band and the antennas in operation and bulk processed by the VLITE astrophysics pipeline which combines standard tasks from both *AIPS* and *Obit*

¹<http://vlite.nrao.edu/>

data reduction software. The result is about 70 images per day with integration times ranging from 10s to 10,000s of seconds and sensitivities ranging from a few Jy to 100s of μ Jy. The imaged field of view ranges from 3 to 50 square degrees with resolutions ranging from a few arcsec to a few arcmin as the VLA cycles through its four array configurations, labeled A, B, C and D array, every four months or so.

2. VDP

The VLITE Database Pipeline, or VDP², is a collection of Python scripts to automate the measurement and database archiving of radio-emitting astronomical sources detected in VLITE images. VDP uses the Python Blob Detector and Source Finder (PyBDSF; Mohan & Rafferty 2015) for source finding and PostgreSQL for database storage. The sky indexing scheme Q3C (Koposov & Bartunov 2006) is utilized to enable high performance queries and positional matching across tables as the VLITE catalog grows.

Data flow through VDP is broken into four stages, shown in Figure 1. These stages can be run in succession on one image at a time in a single execution of the pipeline, or they can be run one stage at a time in multiple executions of the pipeline while processing multiple images each time in each separate stage. The former is the default preferred method since the latter requires multiple executions of the pipeline. Stages can be turned on and off through a YAML configuration file.

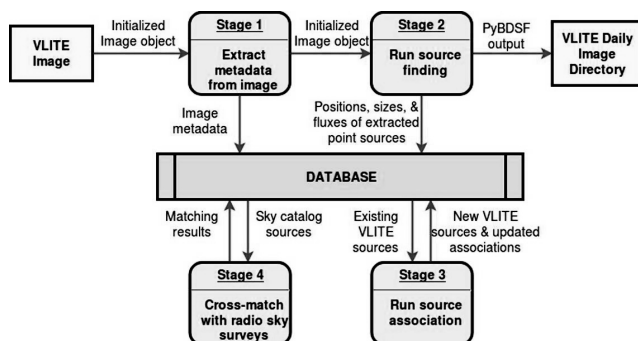


Figure 1. Data flow diagram for the VDP post-processing pipeline. Storage in a queryable database enables straightforward analysis for astrophysical applications and enhances data legacy value.

The first stage of the pipeline simply reads the VLITE image and records some of its header metadata into the database image table. Images are processed in time order based on their Modified Julian Date.

VLITE data is vastly inhomogeneous making quality assurance vitally important to filter out unusable images. The first stage includes quality checks that flag bad images based on their header keywords. If an image fails a quality check, it is assigned an error id corresponding to the failed requirement, its information is added to the image table

²obtainable from <https://github.com/epolisensky/VLITE/>

and the pipeline moves on to the next image. The failed image is not pushed through the remaining quality checks or allowed to pass through to any other stages. The exception is if a known “problem source,” a source known to frequently produce imaging artifacts, is in the field-of-view. These images simply get flagged. Quality checks can be turned off in the configuration file.

In Stage 2, VDP integrates PyBDSF to automate finding and measuring sources in the VLITE images. PyBDSF measures sources by grouping Gaussians that were fit to islands of contiguous pixels. Islands are formed by finding all pixels with a peak flux greater than 5σ and includes all surrounding pixels above 3σ . These default threshold values can be changed in the configuration file. Identification of pixels above the thresholds depends on the local mean and rms. PyBDSF calculates background mean and rms images using a sliding 2-D box with interpolation. The box size and step size in-between are controlled by the rms box parameter.

Experience has shown that it is better to specify the rms box parameter for VLITE images rather than have PyBDSF calculate it internally. Images with bright stripe artifacts will fail miserably with the internal default option. Setting the box size to 1/10th the image size in pixels and the step size to 1/3rd the box size seems to help avoid identifying bright artifacts as real emission while still capturing most of the real sources. The rms box bright parameter is also calculated for every image as 1/5th times the rms box sizes. This parameter is used by PyBDSF only if adaptive rms box is set to True in the configuration file. This tells PyBDSF to use the smaller box size around bright sources where there tend to be more imaging artifacts.

PyBDSF operates on the full VLITE image but sources outside a defined circular field-of-view are removed afterwards. This is done to ensure that cone search queries in the database return sources which lie in the same well-defined field-of-view as the images. The radius of the circular field is half the image size, which for VLITE is 1° for A array, 2° for B, 3° for C and 4° for D. The scale parameter in the VDP configuration file can be used to make the field-of-view radius smaller or larger if retaining sources in the image corners is desired.

Properties of the sources and islands are written to the database detected source and detected island tables, respectively. A ds9 region file is also created for every image. A 1-D primary beam correction factor is applied to all flux measurements from PyBDSF and recorded in the corrected flux table. The applied primary beam correction factor was determined empirically and depends on the source’s distance from the image center, which is also recorded in the corrected flux table, and the primary observing band.

A second round of quality checks are performed on the source finding results before they are inserted into the database tables. Images are flagged if PyBDSF failed to process for any reason or if there were no sources extracted. Any image that takes longer than 5 minutes to process will fail with a timeout error to avoid PyBDSF getting stuck trying to fit Gaussians to large imaging artifacts. We also define a metric to flag images where the number of detected sources is much larger than what is expected based on source counts from survey catalogs and the image’s noise.

Stage 3 is the association stage, it condenses multiple detections of a single source from different images into one entry in the associated source database table. Detections of the same source are required to be at similar spatial resolutions before being associated to avoid differences in source structure, i.e. a resolved double vs. unresolved single. The resolution of an image is defined by the beam semi-minor axis size so

it is less sensitive to elongated beam shapes. Currently, images are divided into four resolution classes roughly corresponding to the four VLA configurations.

In stage 4 all VLITE sources are cross-matched with other radio sky surveys and catalogs to help isolate transient candidates and compare fluxes across the radio spectrum. As for the association stage, cross-matching is restricted between sources with similar spatial resolutions: the resolution of the catalog has to be in the same resolution class as the image. The resolution classes are the same as for association except the first two classes (A & B array) are combined so that there is at least one all-sky survey included.

Execution time mostly depends on the number and size of the images being processed. Typical processing times are 45 – 90 seconds per image for A array, 15 – 45 for B, and 5 – 15 for C & D array, on average. The bottleneck is source finding and measurement with PyBDSF.

3. Uses and Future Development

The VDP database is being used to derive metrics separating point-like and extended sources, to refine the VLITE primary beam corrections for each primary observing band, to investigate the reliability of the VLITE flux scale and calibration, to search for steep spectrum and other exotic sources, and to identify transient candidates.

Development of VDP is ongoing with efforts focused on methods to identify and reduce imaging artifacts, incorporating light curve metrics into the associated source table to enable variable source detection and automate transient source identification, and enhancing the matching capabilities with the goal of cross-matching with surveys at infrared and optical wavelengths.

Acknowledgments. Basic research in radio astronomy at the US Naval Research Laboratory is supported by 6.1 Base Funding. Construction and installation of VLITE was supported by NRL Sustainment Restoration and Maintenance funding. The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

References

- Clarke, T. E., Kassim, N. E., Briskin, W., Helmboldt, J., Peters, W., Ray, P. S., Polisensky, E., & Giacintucci, S. 2016, in *Ground-based and Airborne Telescopes VI*, vol. 9906 of *Proceedings of the SPIE*, 99065B
- Helmboldt, J. F., Kassim, N. E., & Teare, S. W. 2015, *Earth and Space Science*, 2, 387
- Koposov, S., & Bartunov, O. 2006, in *Astronomical Data Analysis Software and Systems XV*, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of *Astronomical Society of the Pacific Conference Series*, 735
- Mohan, N., & Rafferty, D. 2015, *PyBDSF: Python Blob Detection and Source Finder*, *Astrophysics Source Code Library*. 1502.007
- Polisensky, E., Lane, W. M., Hyman, S. D., Kassim, N. E., Giacintucci, S., Clarke, T. E., Cotton, W. D., Cleland, E., & Frail, D. A. 2016, *ApJ*, 832, 60. 1604.00667

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Gaia DR2 and the Virtual Observatory: VO in Operations New Era

Jesús Salgado,¹ Juan González-Núñez^{2,3}, Raúl Gutiérrez-Sánchez⁴,
Juan-Carlos Segovia², Alcione Mora⁵, Jorgo Bakker⁶, Thomas Boch⁷, Mark
Allen⁷, Nigel C. Hambly⁸, Stelios Voutsinas⁸, Markus Demleitner⁹, Gregory
Mantelet^{7,9}, Javier Durán¹⁰, Elene Racero², Pilar de Teodoro¹¹, Deborah
Baines¹, Bruno Merín⁶, and Christophe Arviset⁶

¹*ESAC Science Data Centre, Quasar Science Resources for ESA-ESAC, Spain;
Jesus.Salgado@sciops.esa.int*

²*ESAC Science Data Centre, SERCO for ESA-ESAC, Spain;*

³*ETSE Telecomunicación, Universidade de Vigo, Spain;*

⁴*ESAC Science Data Centre, Telespazio-Vega UK Ltd. for ESA-ESAC, Spain;*

⁵*Gaia SOC, Aurora for ESA-ESAC, Spain;*

⁶*European Space Agency (ESA), Spain;*

⁷*Observatoire Astronomique de Strasbourg, Strasbourg, France;*

⁸*Institute for Astronomy, University of Edinburgh, UK;*

⁹*Astronomisches Rechen-Institut, Zentrum für Astronomie der Universität
Heidelberg, Germany;*

¹⁰*ESAC Science Data Centre, RHEA for ESA-ESAC, Spain;*

¹¹*ESAC Science Data Centre, Aurora for ESA-ESAC, Spain;*

Abstract. During the last decade, the IVOA (International Virtual Observatory Alliance) has been tasked with the difficult task of defining standards to interchange astronomical data. These efforts have been supported by many IVOA partners in general and by the ESAC Science Data Centre (ESDC) in particular, that have been collaborating in the definition of standards and in the development of astronomical VO-inside archives. New ESDC archives, like Gaia, ESASky and the ones in development like Euclid, makes use of VO standards not only as a way to expose the data but, also, as the architectural design of the system.

The Gaia Data Release 2 archive, has been a global effort done not only concentrated into the central archive at ESAC, but also as a collaboration with partner data centers like CDS, ARI, ROE and other members of DPAC. All Gaia partners make use of VO protocols to expose Gaia data as a principle. With this release, the level of dissemination and community endorsement of the VO protocols have entered into a new phase. The percentage of the Gaia expert community that makes use of VO standards has been increased to an unprecedented level of use; 34,000 users accessing the ESA Gaia Archive interface; over 5,000 advanced users sending more than 1,5 million data analysis queries during the only the first week and just counting the ESA Gaia Archive.

These standards are offered to the community in a transparent way (like SAMP or VOSpace to interchange data), as VO protocols extensions like the Tabular Access Pro-

TOCOL extension (TAP+) or as direct use, like the Astronomical Data Access Language (ADQL) that the users learn in order to implement data mining scientific use cases that were almost impossible in the past.

We will describe how VO protocols simplify the work of design and implementation of the astronomical archives and the current level of endorsement by the scientific community.

1. Introduction

The ESDC (Arviset 2015, ESAC Science Data Centre), located at ESAC, is the responsible of the design and implementation of the ESA astrophysical, planetary and heliophysics science archives. This group is also responsible of the long term preservation of the data, so the data is preserved and accessible even long after the finish of the missions.

As part of the work of the Gaia Archive, Gaia DR2 was a challenge due to the publication of a catalog providing full astrometric solution for 1.3 billion stars with the more accurate positions in astronomy (Gaia Collaboration et al. 2018) with an unprecedented impact on the science community. DPAC (Els et al. 2014, Data Processing and Analysis Consortium) is the consortium responsible for the processing of Gaia's data and responsible of the production of the Gaia Catalog, so there are quite close links between DPAC and the ESDC.

2. Gaia Archive

Main archive data is located in ESAC (Salgado et al. 2017) and it is based on extensions of IVOA (International Virtual Observatory Alliance) protocols so it can be considered one of the first pure VO-based operational archives. Main protocol is TAP+, an IVOA TAP compatible service with database schema for the users and table sharing so users can work with their tables at server side.

Also, integration with Python module be done through the well known Astropy distribution (ESDC 2018, `astroquery.gaia`). Query language used is the IVOA Astronomical Data Query Language (ADQL) enabling data exploitation

In order to ensure that the distribution of the data do not have problems during the DR2 event, different copies of the data were provided by DPAC through affiliate centers:

- Centre de Données astronomiques de Strasbourg (CDS - Strasbourg)
- Astronomisches Rechen-Institut (ARI - Heidelberg)
- ASI Space Science Data Center (SSDC - Rome)
- Institut für Astrophysik Potsdam (AIP - Postdam)

These data centers not only provided copies of the data but, also, specific services on the data. ESDC also offered Gaia data through ESASky (Giordano et al. 2018), a pure graphical interface.

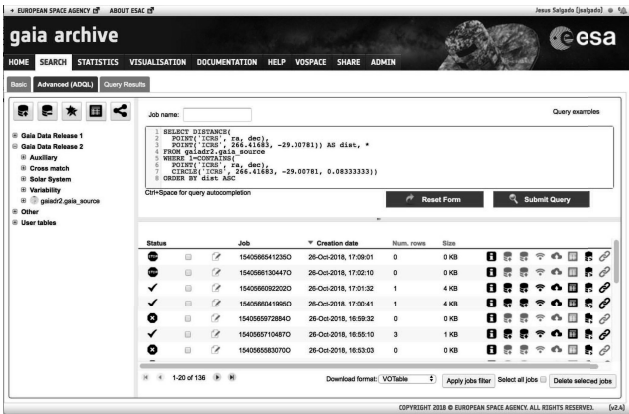


Figure 1. Gaia Archive at ESAC offers the possibility to execute very complex queries on the Gaia catalog so scientific community can do the necessary data mining

3. Gaia DR2 event

Data releases of astronomical data are becoming global events. This is really new as, in the past, only launch events or planetary images obtained this kind of impact on the general media. Many newspapers and televisiones reflected the Gaia DR2 event. In many occasions, digital newspapers also contain links to the science data archive what, in some cases, could be misleading.



Figure 2. A big number of media reflects the Gaia DR2 as a major event

During the first week, the archive received 1.535.752 ADQL queries. Around 34,000 users visited the User Interface. Analyzing the behavior, around 5,000 different users were doing advanced astronomical server side queries, through asynchronous

ADQL and/or creating user database schema. This analysis allows us to make the difference between general non-professional users and scientists doing data exploitation. Queries were received from user interface and, mostly, scripts (Python). Also, a big number of queries were received from VO enabled applications, what justify the creation of services VO compatible. Advanced users number finally obtained was in line with Gaia community size, what implies that most of the scientific community was able to access the Gaia archive to explore the data.

4. Data from DPAC affiliate data centers

Traffic was distributed to the affiliate data centers. In the case of CDS (left figure), users make use of extra services that CDS provides (VizieR, crossmatch services, Aladin, HiPS, etc). The peak on the astronomical community use of CDS services can be clearly see into the stats. Same thing happened at the ARI archive. Apart from an unprecedented use of the IVOA registry hosted at Heidelberg, ARI archive had very big numbers of use. Analysis done (right) shows that most of the queries were received from Python, showing that the new era of data mining on astronomical data is already here.

5. Conclusions

Analysis of the use of the advanced IVOA compliant services for the Gaia archive has proved that these services are being used massively by the scientific community. In fact, the Gaia archive is now a reference for the astronomical community and, as based in VO from its architectural design, it can be considered a different phase of operations for the VO.

Some of the characteristics of the VO protocols have been adapted and extended to fulfill the requirements of an astronomical archive but this experience from the community can be also be used to extend and evolve the protocols themselves, creating a user experience based evolution more than a theoretical one.

References

- Arviset, C. 2015, in Science Operations 2015: Science Data Management, id.2, 2
- Els, S. G., Lock, T., Comoretto, G., Gracia, G., O'Mullane, W., Cheek, N., Vallenari, A., Ordóñez, D., & Beck, M. 2014, in Modeling, Systems Engineering, and Project Management for Astronomy VI, vol. 9150 of Proceedings of the SPIE, 915016
- ESDC 2018, astroquery.gaia, <https://astroquery.readthedocs.io/en/latest/gaia/gaia.html>. [Online; accessed 28-November-2018]
- Gaia Collaboration, Brown, A. G. A., Vallenari, A., Prusti, T., de Bruijne, J. H. J., Babusiaux, C., Bailer-Jones, C. A. L., Biermann, M., Evans, D. W., Eyer, L., & et al. 2018, A&A, 616, A1. 1804.09365
- Giordano, F., Racero, E., Norman, H., Vallés, R., Merín, B., Baines, D., López-Caniego, M., Martí, B. L., de Teodoro, P., Salgado, J., Sarmiento, M. H., Gutiérrez-Sánchez, R., Prieto, R., Lorca, A., Alberola, S., Valtchanov, I., de Marchi, G., Álvarez, R., & Arviset, C. 2018, Astronomy and Computing, 24, 97. 1811.10459
- Salgado, J., González-Núñez, J., Gutiérrez-Sánchez, R., Segovia, J. C., Durán, J., Hernández, J. L., & Arviset, C. 2017, Astronomy and Computing, 21, 22. 1710.10509

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

The OV-GSO Data Center

M. Sanguillon,¹ J.-M. Glorian,² and C. Vastel²

¹*LUPM, Université Montpellier, CNRS/IN2P3, France;*
Presenter at ADASS michele.sanguillon@umontpellier.fr

²*IRAP, Université de Toulouse, CNRS, UPS, CNES, Toulouse, France*

Abstract. The OV-GSO ¹ (Observatoire Virtuel du Grand Sud-Ouest: <https://ov-gso.irap.omp.eu/>) is one of the six French Astrophysical Data Center recognized by INSU/CNRS (National Institute for Earth Sciences and Astronomy) since 2013. It aims at providing, for Astrophysical and Planetology research, data processing, archiving, and dissemination. The OV-GSO gathers five different themes: Sun-Earth (STORMS: Solar Terrestrial ObseRvations and Modeling Service, CLIMSO-DB: database of images from the CLIMSO coronagraphs at Pic du midi observatory in the Pyrénées), planetary plasmas (CDPP: French national data center for natural plasmas of the Solar System), interstellar medium (CASSIS: free interactive spectrum analyzer, CADE: Analysis Center for Extended Data, KIDA: database of kinetic data of interest for astrochemical studies), stellar spectra (PolarBase: database of high resolution spectropolarimetric stellar observations, Pollux: stellar spectra database proposing access to theoretical data), and high energy astrophysics (SCC-XMM: XMM-Newton Survey Science Centre). We present in this article the different services that are hosted at the center, the different tools that have been developed, and the OV standards and protocols that have been used within this Data Center.

1. Introduction

The OV-GSO Data Centre was officially set-up in 2013 after approval by INSU of CNRS. Its role is to support more specific and so-called *reference services* which provide dedicated and community services in relation with relevant astrophysical data. It also promotes and encourages the deployment of Virtual Observatory (VO) techniques, at the regional level. The actual distribution of regional data centers at the national level can be seen in Fig. 1. OV-GSO covers all the open and "science ready" data, and VO-oriented activities of Bordeaux (LAB), Montpellier (LUPM) and Toulouse (IRAP) laboratories for astrophysics. The goal of the OV-GSO is to share common tools for different thematics in astrophysics for a better use of the astrophysical data. This need has revealed to be crucial with the wealth of high-spatial/spectral resolution spectra for the past 15 years with a new generation of ground/space-based observatory. Dealing with a large quantity of data leads to optimized tools and a better and common access to the many databases.

¹<https://ov-gso.irap.omp.eu/>

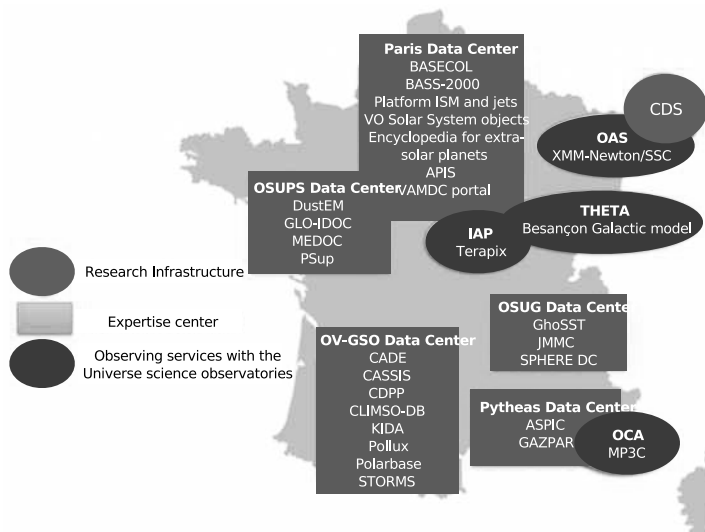


Figure 1. Map of the many data centers for the analysis, archiving and diffusion of astrophysical data in France. OV-GSO covers the whole south-west area.

2. Reference services

2.1. STORMS

STORMS (<https://stormsweb.irap.omp.eu/>), which stands for Solar Terrestrial ObseRvations and Modeling Service, is a public service providing tools and data to perform studies in heliophysics and space weather, and to study and model the influence of solar activity on the geospace environment, as well as on planets or any other solar system bodies (comets, asteroids or spacecrafts). The main tool it provides so far, propagationtool.cdpp.eu, was jointly developed with CDPP. It is meant for the tracking of solar storms, streams and energetic particles in the heliosphere.

2.2. CLIMSO-DB

CLIMSO (<http://climso.irap.omp.eu/>) stands for Christian Latouche IMageur Solaire. CLIMSO is an astronomical observation instrument at Pic du midi observatory in the Pyrénées (France) specialized in the study of the Sun. It makes multiple films of the sun, particularly the globality of the surface and corona. The aim of this instrument is to study the course of the dynamic phenomena in the solar atmosphere by taking into account the great heterogeneity in temperatures, densities, magnetic and electric properties of these areas. It thus acts as a total diagnostic of the solar activity (simultaneously cold corona, hot corona, surface events). It provides images via the French ground solar data BASS2000.

2.3. CDPP

The CDPP (<http://cdpp.irap.omp.eu/>) is the french national data center of expertise concerning terrestrial and planetary plasma data. It was created in 1998 by

CNRS/INSU and the French space agency CNES. It assures the long term preservation of data obtained primarily from instruments built using French resources, and renders them readily accessible and exploitable by the international community. The CDPP also provides services to enable on-line data analysis (AMDA, see amda.cdpp.eu), and 3D data visualization in context (3DView, see 3dview.cdpp.eu). The CDPP also plays an important role in the development of interoperability standards (see e.g., Génot et al. (2014)).

2.4. CASSIS

CASSIS (<http://cassis.irap.omp.eu/>) started in 2005 and is an interactive spectrum analyzer that was originally proposed for the scientific exploitation of (far-infrared and submillimetric) data from the Herschel Space Observatory. CASSIS allows users to visualize observed or synthetic spectra, together with a line identification tool. It can also predict spectra which may be observed by any (single-dish, so far) telescope. Comparison between observations and synthetic spectra is also possible with the same tool (see Vastel et al. (2015)). CASSIS is now evolving towards a multi-purpose spectral analysis tool, operating beyond its initial range of application.

2.5. CADE

CADE (<http://cade.irap.omp.eu/>) is an analysis center for extended data. It provides maps of ancillary astronomical data to users in the astronomical community. The emphasis of CADE is on extended sky emission. The database is provided in the HEALPix sky pixelization scheme. CADE pursues a strategy of ancillary data ingestion in the HEALPix format, generating partial or all-sky HEALPix files from astronomical data that is represented using the more traditional local World Coordinate System (WCS) FITS format (Paradis et al. (2012)). This method guarantees the photometric accuracy of the transformation with minimal data loss during the transformation. CADE provides astronomical data production in the HEALPix format, data archiving and diffusion to the community.

2.6. KIDA

KIDA (<http://kida.obs.u-bordeaux1.fr/>) is a database of kinetic data of interest for astrochemical (interstellar medium and planetary atmospheres) studies. In addition to the available referenced data, KIDA provides recommendations over a number of important reactions. Chemists and physicists can also add their own data to the database. KIDA also distributes a code, named Nahoon, to study the time-dependent gas-phase chemistry of 0D and 1D interstellar sources. Details about the KIDA database can be found in Wakelam et al. (2012).

2.7. PolarBase

PolarBase (<http://polarbase.irap.omp.eu/>) is a database of high resolution spectropolarimetric stellar observations. It was officially opened to the public in 2013. This service distributes high resolution optical stellar spectra from the Espadons at CFHT and Narval at TBL spectropolarimeters. Reduced spectra, in various Stokes parameters, are delivered to the community, as well as standardized extracted polarized signatures. A complete description of the database can be found in Petit et al. (2014).

2.8. Pollux

Pollux (<http://pollux.oreme.org/>) is a stellar spectra database, developed at the Laboratoire Univers et Particules de Montpellier, giving access to theoretical data. For that purpose, high resolution synthetic spectra have been computed using the best available models of atmosphere (CMFGEN, ATLAS and MARCS), high-quality spectral synthesis codes (CMF FLUX, SYNSPEC and TURBOSPECTRUM), atomic line lists from VALD database, and specific molecular line lists for cool stars are provided. Spectral types from O to M are represented for a large set of fundamental parameters: T_{eff} , $\log(g)$, $[Fe/H]$, and specific abundances (Palacios et al. 2010).

2.9. SCC-XMM

The XMM-Newton Survey Science Centre (<http://xmmssc.irap.omp.eu/>) produces catalogs of all X-ray sources detected with the satellite launched in 1999. The SSC has responsibilities within the XMM-Newton project in four main areas: compilation of the Serendipitous Source Catalogue, follow-up/identification program for the serendipitous X-ray sky survey (XID Programme), pipeline processing of all XMM-Newton observations, development of science analysis software for XMM-Newton.

3. Management and operations

The OV-GSO is managed with with a technical resources leader (Jean-Michel Glorian) and a scientific leader (Charlotte Vastel). In December 2018, operations of OV-GSO involve about 7 technical (IT) personnel, and about 18 scientists in the Bordeaux-Toulouse-Montpellier area. The typical annual budget of the data center is about 50 kEuros. One of the major task of OV-GSO is to guarantee the continuity of all services. We also regularly contribute to the various virtual observatories communities, both at the national and international levels (e.g., IVOA). This concerns our recurrent participation to the bi-yearly so-called Interop meetings of the IVOA, as well as propositions of tutorials (e.g., SpecFlow at euro-vo.org scientific tutorials page, or Paletou & Zolotukhin (2014)). Locally, we set-up a monthly dedicated seminar, oriented towards the use and implementation of Virtual Observatory standards and protocols. Our various activities can be followed at ov-gso.irap.omp.eu

References

- Génot, V., André, N., Cecconi, B., & al. 2014, *Astronomy and Computing*, 7, 62
- Palacios, A., Gebran, M., Josselin, E., Martins, F., Plez, B., Belmas, M., & Lébre, A. 2010, *Astronomy and Astrophysics*, 516, 13
- Paletou, F., & Zolotukhin, I. 2014, arXiv:1408.7026
- Paradis, D., Dobashi, K., Shimoikura, T., Kawamura, A., Onishi, T., Fukui, Y., & Bernard, J.-P. 2012, *Astronomy and Astrophysics*, 543, 103
- Petit, P., Louge, T., Théado, S., & al. 2014, *Publications of the Astronomical Society of the Pacific*, 126, 469
- Vastel, C., Bottinelli, S., Caux, E., Glorian, J.-M., & Boiziot, M. 2015, *Proceedings of the Annual meeting of the French Society of Astronomy and Astrophysics*, 313
- Wakelam, V., Herbst, E., Loison, J.-C., & al. 2012, *The Astrophysical Journal Supplement Series*, 199, 21

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

The TESS Science Data Archive

Daryl Swade,¹ Scott Fleming,¹ Jon M. Jenkins,² David W. Latham,³
Edward Morgan,⁴ Susan E. Mullally,¹ and Roland Vanderspek⁴

¹*STScI, Baltimore, MD, USA; swade@stsci.edu*

²*NASA/Ames, Moffett Field, CA, USA*

³*CfA, Cambridge, MA*

⁴*MIT, Cambridge, MA*

Abstract. The Transiting Exoplanet Survey Satellite (TESS) is an all-sky survey mission designed to discover exoplanets around the nearest and brightest stars. The Mikulski Archive for Space Telescopes (MAST) at the Space Telescope Science Institute will serve as the archive for TESS science data. The services provided by MAST for the TESS mission are to store science data and provide an Archive User Interface for data documentation, search, and retrieval. The TESS mission takes advantage of MAST multi-mission architecture to provide a cost-effective archive that allows integration of TESS data with data from other missions.

1. Introduction

The Transiting Exoplanet Survey Satellite (TESS) is a NASA Astrophysics Explorer mission designed to discover exoplanets around the nearest and brightest stars. (Ricker et al. 2015) TESS was launched on April 18, 2018 on a SpaceX Falcon 9 rocket and began science observations in July 2018. The Mikulski Archive for Space Telescopes (MAST) at the Space Telescope Science Institute serves as the archive for TESS science data. (Swade et al. 2018)

2. Science Objectives

TESS will search for transiting planets by conducting large area surveys of bright stars and known M dwarfs within 60 parsecs. Planetary host stars discovered by TESS will require follow-up observations with ground and space-based observatories, such as JWST, in order to further characterize the exoplanets.

The TESS mission will generate valuable science data products for exoplanet and other astronomical studies. Target pixels and associated light curves are sampled every two minutes for approximately 15,000 stars per sector. Full frame images (FFI) contain 24 x 96 degree areas of the sky sampled every 30 minutes.

3. Observatory Operations Concept

The TESS observatory includes four cameras. Each camera consists of f/1.4 lens with an effective aperture of 10.5 cm and a 24x24 degree field of view. Each camera field of view is imaged onto four CCDs, for a total of 16 CCDs in the focal planes. The individual pixel size is 21 arc-seconds on the sky.

Target stars are selected from the TESS Input Catalog (TIC)(Stassun et al. 2018). At present, the TIC contains approximately half a billion persistent luminous objects over the entire sky that are potential two-minute targets or are needed to document nearby fainter stars that contaminate the target photometry.

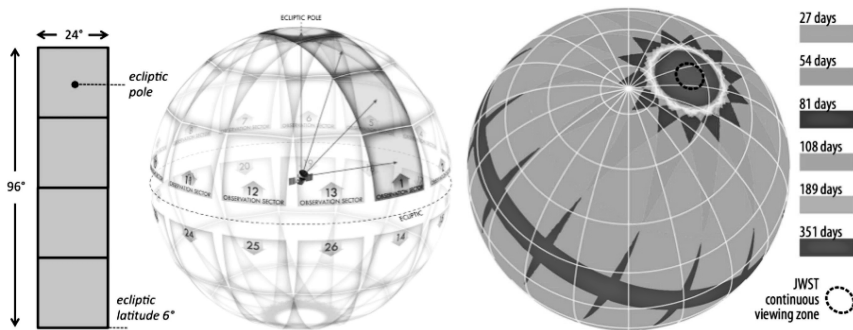


Figure 1. TESS camera field-of-view and sector mapping (Ricker et al. 2015)

As shown in Figure 1, the four TESS cameras cover a field-of-view of 24 degrees x 96 degrees, raised by about 6 degrees from the ecliptic. Thirteen sectors are needed to cover one hemisphere. As declination increases, there is an increasing amount of overlap with the field-of-view. Near the ecliptic poles sources can be observed continuously.

TESS has an Earth-centered orbit as shown in Figure 2. The TESS orbit is designed for continuous periods of science observation, yet a perigee sufficiently close to Earth to allow the downlink of an entire orbit of data over Ka-band. The orbit, named P/2, is a 13.7-day Earth-centered orbit in 2:1 resonance with the Moon. The P/2 mission orbit has an apogee of 59 Earth radii and a perigee of 17 Earth radii. There are two TESS orbits per sector.

As shown in Figure 3 from the TESS Observatory Guide¹, CCD detector 2-second integrations are summed into 2-minute postage stamps around approximately 15,000 target stars per sector and 30-minute full frame images.

Table 1 lists the science data products that are significant components of the data volume.

¹<https://heasarc.gsfc.nasa.gov/docs/teess/docs/>

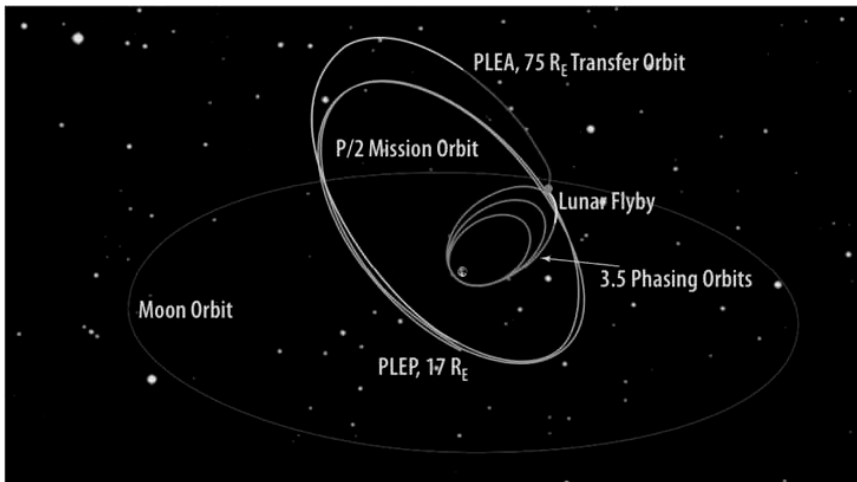


Figure 2. TESS orbit (Ricker et al. 2015)

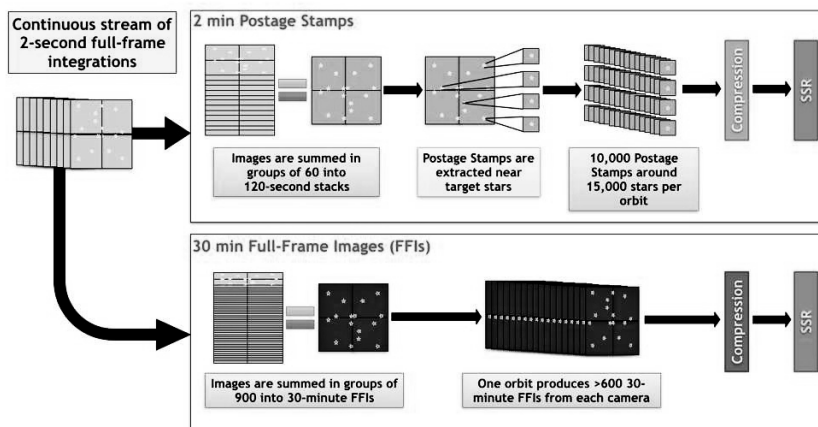


Figure 3. Instrument science data flow.

4. Archive User Interface

Archive users will access TESS data files and catalogs through the MAST web-based Archive User Interface (AUI) ². The AUI allows users to search, query, preview, and

²<http://archive.stsci.edu>

Table 1. Science data products that are significant components of the data volume.

Data Type	Data Volume (GB/sector)	Data Volume (GB/year)	Notes
DSN IDRs	300	3900	S/c SFDU data for safe keeping, compressed
Uncalibrated FFIIs	380	5000	16 files per 30 minute cadence
Calibrated FFIIs	770	10000	16 files per 30 minute cadence
Target pixel files	760	9800	15000 targets/month, 100 pixels/target, 2 minute cadence
Light curves	30	410	One per target per sector
Collateral pixel files	890	11500	Leading virtual columns, trailing virtual columns, smear row, and virtual row
Total	3100	40600	

retrieve data. The TESS archive user experience is integrated into the MAST user interface along with data from other missions such as Kepler, HST, and JWST.

The MAST AUI provides a TESS mission specific home page³. The TESS specific page provides access to TESS mission data and documentation. The page also provides links to MAST tools that can be applied to TESS data.

References

Ricker, G. R., Winn, J. N., Vanderspek, R., Latham, D. W., Bakos, G. Á., Bean, J. L., Berta-Thompson, Z. K., Brown, T. M., Buchhave, L., Butler, N. R., Butler, R. P., Chaplin, W. J., Charbonneau, D., Christensen-Dalsgaard, J., Clampin, M., Deming, D., Doty, J., De Lee, N., Dressing, C., Dunham, E. W., Endl, M., Fressin, F., Ge, J., Henning, T., Holman, M. J., Howard, A. W., Ida, S., Jenkins, J. M., Jernigan, G., Johnson, J. A., Kaltenegger, L., Kawai, N., Kjeldsen, H., Laughlin, G., Levine, A. M., Lin, D., Lissauer, J. J., MacQueen, P., Marcy, G., McCullough, P. R., Morton, T. D., Narita, N., Paegert, M., Palte, E., Pepe, F., Pepper, J., Quirrenbach, A., Rinehart, S. A., Sasselov, D., Sato, B., Seager, S., Sozzetti, A., Stassun, K. G., Sullivan, P., Szentgyorgyi, A., Torres, G., Udry, S., & Villaseñor, J. 2015, *Journal of Astronomical Telescopes, Instruments, and Systems*, 1, 014003

Stassun, K. G., Oelkers, R. J., Pepper, J., Paegert, M., De Lee, N., Torres, G., Latham, D. W., Charpinet, S., Dressing, C. D., Huber, D., Kane, S. R., Lépine, S., Mann, A., Muirhead, P. S., Rojas-Ayala, B., Silvotti, R., Fleming, S. W., Levine, A., & Plavchan, P. 2018, *AJ*, 156, 102

Swade, D., Fleming, S., Jenkins, J. M., Latham, D. W., Morgan, E., Mullally, S. E., Sparks, W., & Vanderspek, R. 2018, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 10704, 1070415

³ <http://archive.stsci.edu/tess/>

Session IX

Software for Solar System Astronomy

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

ESASky: A New Window for Solar System Data Exploration

Elena Racero,¹ Fabrizio Giordano,¹ Benoit Carry,² Jerome Berthier,³
 Juan Gonzalez,¹ Henrik Norman,¹ Deborah Baines,¹ Bruno Merin,¹
 Belen Lopez Marti,¹ Marcos Lopez-Caniego,¹ Pilar de Teodoro,¹ Jesus
 Salgado,¹ Christophe Arviset¹

¹*European Space Astronomy Centre (ESA-ESAC), Madrid, Spain*
eracero@sciops.esa.int

²*Observatoire de la Cote d’Azur (OCA), Nice, France*

³*Observatoire de Paris (IMCCE), Paris, France*

Abstract. We present here the first integration of the search mechanism for solar system objects through ESASky. Based on the IMCCE Eproc software for ephemeris pre-computation, it allows fast discovery of photometry observations from ESA astronomical missions that potentially contain these objects within their field of view. In this first integration, the user can input a target name and retrieve on-the-fly the results for all the observations that match the input provided, that is, that contains within the exposure time frame the ephemerides of such objects. At the moment the search mechanism provides access to three major ESA missions, XMM-Newton, Hubble Space Telescope and Herschel Space Observatory, to be extended at a later stage to other relevant data assets.

1. Introduction

Allowing the solar system community fast and easy access to the astronomical data archives is a long-standing issue. Moreover, the everyday increasing amount of archival data coming from a variety of facilities, both from ground-based telescopes and space missions, leads to the need for single points of entry for exploration purposes.

Efforts to tackle this issue are already in place, such as the Solar System Object Image Search by the Canadian Astronomy Data Centre (CADC)¹, plus a number of ephemeris services, such as Horizons (NASA-JPL)², Miriade (IMCCE)³ or the Minor Planet & Comet Ephemeris Service (MPC)⁴.

Within this context, the ESAC Science Data Centre (ESDC), located at the European Space Astronomy Centre (ESAC) has developed ESASky (Giordano et al. 2018),

¹<http://www.cadc-ccda.hia-ihp.nrc-cnrc.gc.ca/en/ssois/>

²<https://ssd.jpl.nasa.gov/horizons.cgi>

³<http://vo.imcce.fr/webservices/miriade/>

⁴<https://www.minorplanetcenter.net/iau/mpc.html>

a science driven discovery portal to explore the multi-wavelength sky providing a fast and intuitive access to all ESA astronomy archive holdings. Released in May 2016, ESASky⁵ is a new web application that sits on top of ESAC hosted archives, with the goal of serving as an interface to all high-level science products generated by ESA astronomy missions. The data spans from radio to x-ray and gamma-ray regimes, and includes the Planck, Herschel, ISO, HST, XMM-Newton and INTEGRAL missions.

In this first integration we enable the user to input a target name, resolved against the SsODNet⁶ service, and retrieve the pre-computed results for all the observations matching the input Solar System Object (SSO) provided. These results are computed over a geometrical cross-match between the observation footprints and the ephemerides of the SSOs within the exposure time frame of the observations.

2. Processing Pipeline

The processing pipeline to pre-compute the output list of potential detections has been developed with the goal of reducing the cardinality of the number of geometrical cross-matches needed, minimising the time required for the computation of the results. For the development, Java and Java Threads have been used on top of the ESAC Grid infrastructure. The workflow can be divided in three main steps, described in this section.

2.1. Ephemerides Sampling

A first even sampling of the apparent position of the SSO from each satellite point of view is performed and stored locally for a fixed time interval of 10 days. This computation is performed with Eproc v3.2 suite provided by IMCCE (Berthier 1998). The time span of this sampling is linked to the life-time of each mission.

The input orbital parameters for asteroids are retrieved from the Lowell Observatory Asteroid Orbital Parameter table⁷. In the comet case, the input is provided by the IMCCE⁸.

To take into account the satellite reference frame for the ephemerides computation, Eproc software can be provided, via the SPICE kernels⁹, with the orbital information for each mission, thus the total time period in the ephemerides sampling is limited to the availability of this information within these kernel files. The table below summarizes the status of each of the kernels used currently by the processing pipeline.

In the case of the Hubble Space Telescope, this file was publicly available at NASA's Navigation and Ancillary Information Facility (NAIF)¹⁰, whereas the XMM-Newton kernel was provided by the Science Operations Centre (SOC) at ESAC. Finally, the Herschel Orbital Element Message (OEM) was produced by the SOC and converted in-house at the ESDC into the appropriate SPICE kernel.

⁵<http://sky.esa.int>

⁶<http://vo.imcce.fr/webservices/ssodnet/>

⁷<http://asteroid.lowell.edu>

⁸<http://www.imcce.fr/en/ephemerides/donnees/comets/index.html>

⁹<https://naif.jpl.nasa.gov/naif/data.html>

¹⁰<http://naif.jpl.nasa.gov/pub/naif/HST/>

Table 1. SPICE Kernel Summary

Satellite	Source	Time Span
Hubble Space Telescope	NAIF	1990/04/26 - 2016/04/22
XMM-Newton	XMM-Newton SOC	1999/12/17 - 2016-08-09
Herschel	Herschel SOC and ESDC	2009/05/16 - 2013/07/01

2.2. Cardinality Reduction

This second step serves as a fast selection of the potential detection candidates per mission dataset, reducing the number of geometrical cross-matches needed in the subsequent step, very costly in terms of CPU times and resources. The position and uncertainty of each object coming from the previous orbit sampling is cross-matched against the selected datasets imaging footprints.

To speed this selection, this process is based on the HEALPix(Górski et al. 2005) tessellation of the sky. HEALPix indexes are used to represent both the sky-path of each object during each time sample and the observation footprint (representation of the FOV of a given instrument). The selection of the HEALPix order is computed based on the minimum distance to the object and its maximum apparent proper motion (see Fig1).

2.3. Geometrical Cross-Match

The output list of candidate observations per SSO undergo then a new precise geometrical cross-match where the position of each object is re-computed for the exact start time and duration of the observation and the cross-match is performed against the observation footprint (Fig.1).

There are three possible scenarios or cross-match types included in this geometrical cross-match:

- The position of the SSO lies within the observation footprint (Fig.1).
- The position of the SSO is not included in the observation footprint, but the uncertainty of the position overlaps with the footprint polygon or contains the footprint polygon.
- None of the SSO positions (start time, end time) nor their uncertainties overlap with a footprint, but the path followed from the start to the end position does cross the footprint polygon.

3. Conclusions

In this contribution we introduced the ongoing work on the first Solar System Object Search Service in the context of ESA astronomy archives. This service is provided through the ESASky application, and aims at enabling users to search for all the potential detections of a Solar System Object (asteroids and comets) within the astronomy imaging observations hosted at ESDC, both targeted and serendipitous. For this

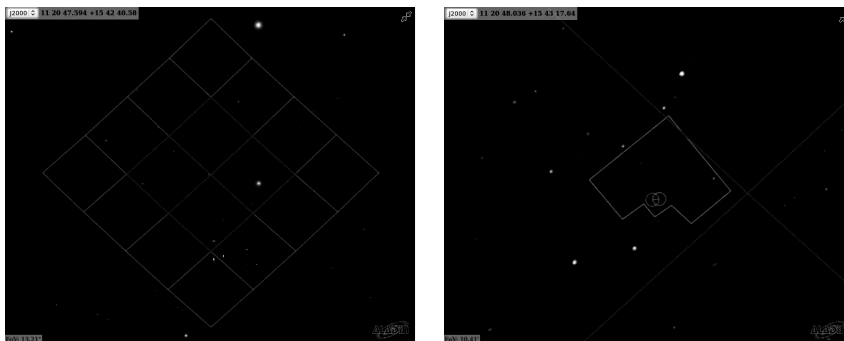


Figure 1. Example of the final two steps of the processing pipeline for comet 67P against HST observation u8ls0304m. *Left*: The cardinality reduction based on the comparison of the HEALPix pixels representing the maximum position uncertainty of 67P in days (green tiles), and the HEALpix cells representing the observation footprint(blue tiles). *Right*: A zoom-in of the area where the final geometrical cross-match step is displayed. Green circles represent the SSO recomputed position and uncertainties (exposure start/end times). In red is the HST observation footprint

first integration, three representative missions covering a wide range of the electromagnetic spectrum, from X-Rays (XMM-Newton) to Far-Infrared (Herschel) including the UV-Near Infrared band from the Hubble Space Telescope, were chosen as a prove of concept.

Future developments will allow user-defined orbital parameters as input and on-the-fly computation of potential detections per object and mission.

References

- Berthier, J. 1998, Notes Scientifiques et Techniques du Bureau des Longitudes, 62, 114
 Giordano, F., Racero, E., Norman, H., et al. 2018, Astronomy and Computing, 24, 97
 Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, Astrophysical Journal, 622, 759

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

The PDS Approach to Science Data Quality Assurance

Anne Raugh

University of Maryland, College Park, Maryland, USA; araugh@umd.edu

Abstract. The Planetary Data System (PDS) was established by NASA in the early 1980s with the mandate not merely to preserve the bytes returned by its planetary spacecraft, but to ensure those data would be available to and usable by future generations. When PDS accepts data for archiving, it must be complete, thoroughly documented, and as far as possible autonomous within the archive (that is, everything needed to understand and use the data must be in the archive as well). In order to maintain usability, PDS must first establish usability of each incoming data submission. The two primary quality assurance tools applied to archive submissions are the PDS4 Information Model and the mandatory External Peer Review. The Information Model guides data preparers to producing well-formatted, well-documented data products that are programmatically accessible, while the External Peer Review ensures the archive submission is complete, usable, and of sufficient quality to merit permanent preservation and support as part of the Planetary Data System archives.

1. Introduction

The Planetary Data System (PDS) was established in response to a "perception that data problems are pervasive throughout space sciences" (CODMAC 1982), and a subsequent Planetary Data Workshop convened at Goddard Space Flight Center because it was "noted with increasing alarm by many in the science community that valuable data sets are disappearing" (Kieffer 1984). NASA charged PDS with preserving and maintaining the usability of planetary mission data in perpetuity; PDS became NASA's guarantee of return on investment in planetary science. As such, when PDS accepts data for archiving the question of quality assurance is primary. In the PDS context, "data quality" is interpreted as meaning data that are complete, thoroughly documented, and compliant with PDS4 data formats and metadata standards. The primary tools for ensuring data quality are the PDS4 Information Model and the External Peer Review. The Information Model guides data preparers to producing well-formatted, well-documented data products using a standardized metadata system that can be programmatically validated, while the External Peer Review ensures the archive submission is scientifically complete, usable, and of sufficient quality to merit permanent preservation and support as part of the Planetary Data System archives.

2. The PDS4 Information Mode (IM)

The PDS4 Information Model (IM) codifies metadata not just for structure, but for provenance, interpretation, and analysis. The XML document structures defined for the

current implementation of the model and its various constituent namespaces establish minimum requirements and present best practices for describing all these aspects of the archival data. The schematic enforcement of these requirements provides a simple, automated approach to ensuring the metadata are present and well-formed.

2.1. Content Management

The foundation layer of the PDS4 IM defines the common metadata types and structures that are used for identification, provenance, and data structure definition in the core, and for all metadata defined in discipline and mission namespaces that extend the IM (Rough & Hughes 2015). Metadata is categorized and organized into dependent groups that are included or not depending on context and observational data content.

The hierarchical organization into classes and subclasses allows metadata to be viewed and used as functional groups. For example, the metadata needed to cite a data product reside in a class contained in an "identification area," which may also contain classes documenting modification history. The hierarchy provides two main benefits: First, the structural organization itself imposes existential constraints for required attributes, units of measure, and so on - to ensure at least a minimal level of metadata is supplied; and second, the full structure as documented in the defining XML schemas provides a complete template to follow for those preparing data for archiving, which helps ensure consistency in what metadata are included and where. The overarching hierarchy that divides the PDS4 IM itself into namespaces extends these templates to cover entire disciplines - providing a standard model for, for example, defining common geometric values related to the observation.

2.2. Schematic Validation

Because the IM is expressed as a combination of XML Schema Definition Language (XSD) Schema, which enforce the structural and hierarchical constraints, and Schematron files, which enforce more complex dependencies related to metadata values and co-dependencies, a broad range of validation can be performed entirely mechanically. More importantly, the canonical validator (the PDS-produced *Validate Tool*) can be deployed to any data preparer's environment and used locally to ensure compliance prior to submission for review.

3. External Peer Review

The PDS External Peer Review is required for all candidate data submissions prior to acceptance for archiving. Equivalent to the refereeing process for journal articles, the PDS External Peer Review presents the candidate data to discipline experts unaffiliated with the creators of the data. These reviewers exercise the data in its archival form by reproducing published results, doing comparative analysis between the candidate data and similar or correlated observations, and so on, using only the archival resources. These reviewers then determine if the data are of archival quality and, where needed, formulate a list of corrections and additions required prior to archiving.

3.1. The Panel

External peer reviewers are chosen to be discipline experts in the data to be reviewed, but must not have been involved in the data preparation process for the data they are

reviewing - neither as a member of the team that took the data or produced the archive, nor as one of the PDS consultants who advised the data preparation team. In general, two independent discipline experts are asked to review the candidate data. PDS personnel do, of course, also validate standards compliance.

3.2. The Process

The typical review takes about two calendar months of time. The data are submitted to PDS, who will do a standards compliance check to ensure that tools that recognize PDS4 format will not encounter problems reading the data. The reviewers (previously chosen) then have one month to exercise the data. They are encouraged to perform some reasonable analytical process - something an end-user might do. Typical analyses include reproducing a published result, correlating the properties of the reviewed data set against another data set, or verifying calibration by calibrating a raw data set to compare to its reduced counterpart.

Reviewers also read accompanying documents, and will generally check metadata for consistency with expected standards. Those familiar with the SPICE toolkit, for example, will frequently check the observational geometry included in the label against the results produced by the toolkit.

At the end of the month, a review meeting is held. The reviewers present their findings, along with proposed revisions that reviewers consider necessary prior to archiving. Data preparers are invited to participate, in order to ask and answer questions. The goal is to produce a specific list of revisions, referred to as a list of "liens", to be applied prior to final delivery, and to ensure that the data provider can address those revisions quickly and completely.

3.3. Revision and Archiving

Following the review, the data preparer typically takes a few weeks to perhaps a few months (where pipeline software must be revised and worked through configuration control) to make the revisions and to submit the final dataset. PDS will do an acceptance review to ensure both standards compliance and resolution of the liens list. Once accepted, the data are posted as archived.

4. Conclusion

The PDS4 Information Model, by requiring a minimum set of metadata, providing templates for data providers to follow in designing metadata for their products, and facilitating rigorous schematic validation, provides quality assurance for metadata syntax and content as well as data structure compliance (via the metadata constraints on structural descriptions). The PDS External Peer Review ensures the data are usable by recruiting field experts to literally use the data, and by declining to accept data for archiving until its usability has been thus demonstrated. The IM and the External Peer Review work together to ensure that data are well-described and usable when they enter the archive.

References

- CODMAC 1982, Data Management and Computation, Volume 1: Issues and Recommendations, Tech. rep., National Academy Press, Washington, D.C. (CODMAC is the Committee on Data Management and Computation, Space Science Board, Assembly of Mathematical and Physical Sciences, National Research Council)
- Kieffer, H. H. 1984, in NASA Conference Publication 2343 (NASA Scientific and Technical Information Branch), 1
- Rough, A. C., & Hughes, J. S. 2015, in AAS/Division for Planetary Sciences Meeting Abstracts #47, AAS/Division for Planetary Sciences Meeting Abstracts, 312.04



(from left to right) Elizabeth Warner, Anne Rough, Gerbs Bauer and Mike Kelley, defending the planet(s). (Photo: Peter Teuben)

Modeling Effects of Stellar UV-Driven Photochemistry on the Transit Spectra of Moist Rocky Atmospheres Around M Dwarfs

Mahmuda Afrin Badhan,^{1,2} Eric T. Wolf,³ Ravi K. Kopparapu,² Giada Arney,² Eliza M.-R. Kempton,¹ Drake Deming,¹ and Shawn Domagal-Goldman²

¹University of Maryland College Park, College Park, MD, USA;
afrin20m@astro.umd.edu

²NASA Goddard Space Flight Center, Greenbelt, MD, USA

³University of Boulder Colorado, Boulder, Colorado, USA

Abstract. 3-D climate modeling has shown that tidally-locked terrestrial planets, at the inner habitable zone edge (IHZ) of M dwarf stars with $T_{eff} > 3000$ K, are able to retain a "moist" atmosphere (i.e. a water vapor rich stratosphere). However, flaring M dwarfs have strong UV activity, which may photodissociate this H₂O. Here, we employ a 1-D photochemical model with varied stellar UV, to assess whether H₂O loss driven by high stellar UV would affect H₂O detectability in *JWST* transmission spectroscopy. Temperature and water vapor profiles are taken from published 3-D climate model simulations of an IHZ Earth-sized planet around a 3300 K M dwarf with an N₂-H₂O atmosphere; they serve as self-consistent inputs for the 1-D model. We explore additional chemical complexity within the 1-D model by introducing other atmospheric species. In this paper, we review our methodology, focusing on the 1-D photochemical model.

1. Introduction

Planets orbiting close enough to their host M dwarf stars to be tidally-locked have their day-sides continuously bombarded by the energetic ultra-violet (UV) stellar irradiation from the star's flare activity. The first habitable zone (HZ) exoplanets to have their atmospheres characterized will likely be such planets orbiting nearby M dwarfs. Observed spectroscopic signatures from transit measurements can identify radiatively active species in a planet's atmosphere. The *James Webb Space Telescope* (*JWST*) should help us constrain exoplanet atmospheric compositions with unprecedented accuracy.

Transit observations sense planetary stratospheres. 3-D climate simulations have shown thick substellar water clouds persist on synchronously rotating HZ planets due to strong vertical convection (Yang et al. 2013). Kopparapu et al. (2017) found that slow rotators around model M dwarf stars (i.e., stellar data without chromospheric emission: UV activity is at blackbody level only) maintain moist-greenhouse conditions (stratospheric H₂O $> 10^{-3}$, Kasting et al. 1993) despite relatively mild surface temperatures (~ 280 K). We may expect to observe stronger H₂O features in the transmission spectra of habitable slow rotators around M dwarfs compared to a true Earth-twin ($f_{H_2O} \sim 10^{-6}$).

However, high UV instellation (i.e., stellar irradiation) from flaring M dwarfs would cause photochemical breakdown of molecular species in the upper atmosphere. This may alter abundances of key gases at altitudes probed by our space IR instru-

ments. The Question: Can the H_2O lost from photodissociation be significant enough at altitudes probed by transmission spectroscopy to affect its detection via *JWST*?

To answer this, we study the composition of a simulated planet within the moist greenhouse regime of Kopparapu et al. (2017), with a 1-D atmospheric model that includes equilibrium chemistry, photochemistry and vertical mixing (i.e. processes that drive the 1-D evolution of atmospheric volatiles). We consider an N_2 - H_2O -rich inner HZ rocky planet modeled around a 3300 K M dwarf, with synthetically varied stellar UV emissions from 1216-4000 Å. We explore the influence of UV activity on the composition of the planetary terminator, which is sensed by primary transit observations.

2. Overview of Method: The 3-D to 1-D "pseudo" Coupling Process

We use the 3-D global climate model (GCM) result of a particular planet-star pair from Kopparapu et al. (2017) as inputs to our models. Specifically, we use terminator mean vertical profiles of P, T, N_2 and H_2O from the GCM as inputs for a 1-D photochemistry model. The GCM is the Community Atmosphere Model v.4. CAM was created by the National Center for Atmospheric Research (NCAR) to simulate the climate of Earth (Neale et al. 2010). To model the 1-D chemical evolution of the atmosphere, we use the Atmos photochemical modeling tool, developed by our group at NASA GSFC.

We augment our 1-D modeled atmospheres with other species and let the atmosphere evolve for five different UV activity profiles, including the original UV-quiet model star used in Kopparapu et al. (2017). The other four are synthetic "active" UV scenarios, created with UV profiles from the MUSCLES Treasury Survey (Youngblood et al. 2016) and Atmos' own spectral database (Domagal-Goldman et al. 2014). We run a single 1-D model per UV case; we determine steady state abundances for all modeled species for a total of five simulated atmospheres. Finally, we use the Exo-Transmit spectral calculation tool (Kempton et al. 2017) to compute the transmission spectra.

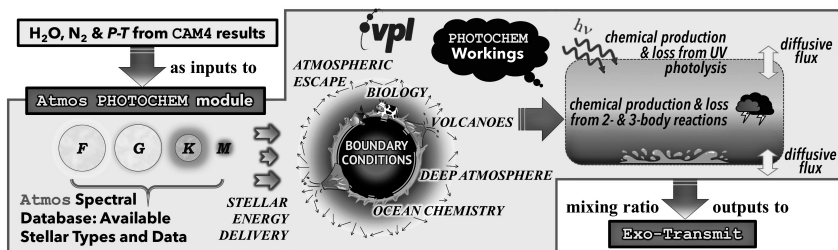


Figure 1. Our 3-D to 1-D "pseudo" coupling method summarized and inner workings of the Atmos 1-D photochemistry model shown.

We have chosen to focus the rest of this paper on the 1-D model's architecture. Further details on the 3-D model, the spectral model, our choice of parameters, and the UV data creation process are covered in Afrin Badhan et al. 2018 (ApJ, in review).

3. 1-D Abundances of Gas Phase Species via the Atmos Photochemical Model

The Virtual Planetary Laboratory's (VPL) Atmos is a coupled 1-D photochemical (PHOTOCHEM) & radiative transfer/convection model (CLIMA) (Arney et al. 2017). Atmos has

been used in 1-D photochemical and climate modeling of early Mars, the Archean and modern Earth atmospheres, and the templates for these validated simulations are part of our public version¹. The two modules can be used standalone; since we do not vary the 1-D P - T profile in our calculations, we only employ the PHOTOCHEM module for this study. The modules are also developed independently; the PHOTOCHEM module can be scaled up for larger planets and to a wider range of densities, pressures and temperatures. Thus, although Atmos was originally written for studying terrestrial worlds only, we are in the process of extending the validity of the PHOTOCHEM to other regimes; we have templates in progress for Hot Jupiters and Saturn's moon Titan. As part of related efforts, we have been updating our stellar database (e.g., for this work), aerosol and haze production mechanisms, reaction pathways and photolysis coefficients.

Using the reverse Euler method, PHOTOCHEM solves a set of nonlinear, coupled ordinary differential equations for the mixing ratios of all species at all heights.

$$\text{Continuity: } \frac{\partial n_i}{\partial t} = P_i - \ell_i n_i - \frac{\partial \Phi_i}{\partial z},$$

$$\text{Flux: } \Phi_i = -K_{zz} n \frac{\partial f_i}{\partial z} - D_i n_i \left[\frac{1}{n_i} \frac{\partial n_i}{\partial z} + \frac{1}{H_i} + \frac{1 + \alpha_{Ti}}{T} \frac{\partial T}{\partial z} \right],$$

where, for species i , n_i = number density (molecules/cm³), P_i = chemical production rate (molecules/cm³/s), ℓ_i = chemical loss frequency (s⁻¹), Φ_i = flux, $f_i = n_i/n$ = mixing ratio ($n = \sum n_i$), K_{zz} = eddy diffusion coefficient (cm² s⁻¹), D_i and α_{Ti} = diffusion coefficient and thermal diffusion coefficient, respectively, of species i with respect to the background. $H_i = k_B T / m_i g$ = scale heights of species i , where k_B = Boltzmann's constant, m_i = its molecular mass, and μ = mean molecular mass of the atmosphere.

Quoting from our ApJ manuscript in review, the method is first order in time and uses second-order centered finite differences in space. The system of equations is explicitly formulated as time-dependent equations that are solved implicitly by a time-marching algorithm. The model is run to steady state for the final mixing ratio profiles. In each step, the model measures the relative change of the concentration of each species per layer of the atmosphere. When all species in all layers change concentrations by < 15% in the time step, the size of the time step grows. When this size exceeds 10¹⁷ seconds (~3 billion years), the model is considered to have attained the steady state (convergence). This means that the resulting solutions have stable chemical profiles on timescales of billions of years, assuming boundary conditions remain the same.

The constant boundary conditions can include biological gas fluxes, volcanic outgassing, atmospheric escape, deep atmosphere abundances (from equilibrium chemistry), and parametrization for ocean chemistry (see Figure 1). Starting boundary conditions to the model can be supplied in the following forms for each modeled species: a) fixed surface deposition efficiency (ν_{dep}), b) constant mixing ratio throughout the atmospheric column (CO₂ in this study), c) fixed mixing ratio at the surface (e.g., N₂ here), or d) constant upward flux ("flux"); first three quantities are dimensionless, fluxes are in molecules/cm²/s. We define H₂ by both ν_{dep} and a vertically distributed upward flux over a user-controlled km range from the surface.

We use the reactions and species list of the "Modern Earth" template in Atmos for these runs. We have 193 forward chemical reactions and 40 photolysis reactions

¹Public Atmos: <https://github.com/VirtualPlanetaryLaboratory/atmos/>

for 40 long-lived and 9 short-lived species made from H, C, O, N, and S, 23 of which participate in photolysis. All other species apart from CO₂ are allowed to vary. We keep the S-based species from the Modern Earth template's list to assure convergence, but assign them extremely low arbitrary boundary values to minimize their presence.

We modify the boundary conditions to simulate an abiotic planet after Harman et al. (2015). In all five runs, we fix CH₄ *flux* to a 1-Earth mass planet abiotic production rate of 1×10^8 molecules/cm²/s (Guzmán-Marmolejo et al. 2013). To determine the lower boundary conditions for a few other varying species, we assume both the atmosphere and ocean obey redox balance (i.e., free electrons are conserved), using the methodology in Harman et al. (2015). The amount of reducing material flowing out of the ocean into the atmosphere can be controlled by using a H₂ *flux* value that balances the amount of oxidants. Thus, we vary the H₂ *flux* across each of the five scenarios until global redox balance is achieved for each case. Conserving redox balance ensures that the ocean composition, and thus the atmospheric concentrations computed, are sustainable over geological timescales with only geological (and not biological) fluxes. H₂O is the only non-background species with mixing ratios provided by the GCM. So, instead of defining surface conditions, we fix the H₂O abundance profile below the tropopause to the mixing ratios of those levels from the GCM. We assume a hydrostatic atmosphere with no top-of-atmosphere (TOA) escape occurring. Escape should primarily affect the long term evolution of the atmosphere and not our computed steady state composition.

Chemistry and vertical transport are both considered for long-lived species. Transport is not factored for short-lived species (e.g. O¹D). Vertical transport within the atmosphere is approximated with K_{zz} (P) coefficients. Vertical mixing has been shown to be much stronger (than Earth) for Earth-sized slow rotators, owing to the strong sub-stellar convection. Thus for our runs, we adopt a constant eddy profile by iteratively determining a single eddy coefficient that allows the input H₂O value at 1 mbar (GCM model's TOA) to be maintained, while letting the atmospheric column above vary.

4. Regarding Results and Future Work

Results of our modeling work and implications will be available in our ApJ paper in review. We are presently also working on a follow-up paper where we focus on varying the boundary conditions of some other species, and see if (and how) they impact trace species, particularly key but hard to observe biosignatures.

References

- Arney, G. N., et al. 2017, ApJ, 836, 49. 1702.02994
 Domagal-Goldman, S. D., et al. 2014, ApJ, 792, 90. 1407.2622
 Guzmán-Marmolejo, A., et al. 2013, Astrobiology, 13, 550
 Harman, C. E., et al. 2015, ApJ, 812, 137. 1509.07863
 Kasting, J. F., et al. 1993, Icarus, 101, 108
 Kempton, E. M.-R., et al. 2017, PASP, 129, 044402. 1611.03871
 Kopparapu, R. K., et al. 2017, ApJ, 845, 5. 1705.10362
 Neale, R. B., et al. 2010, Description of the near community atmosphere model (cam 5.0), ncar/tn-486+str near technical note
 Yang, J., et al. 2013, ApJ, 771, L45. 1307.0515
 Youngblood, A., et al. 2016, ApJ, 824, 101. 1604.01032

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

ZChecker: Finding Cometary Outbursts with the Zwicky Transient Facility

Michael S. P. Kelley,¹ Dennis Bodewits,² Quanzhi Ye,^{3,4} Russ R. Laher,⁴
Frank J. Masci,⁴ Serge Monkewitz,⁴ Reed Riddle,⁴ Ben Rusholme,⁴
David L. Shupe,⁴ and Maayane T. Soumagnac⁵

¹*Department of Astronomy, University of Maryland, College Park, MD 20742, USA* msk@astro.umd.edu

²*Physics Department, Auburn University, Auburn, AL 36849, USA*

³*Division of Physics, Mathematics and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA*

⁴*Infrared Processing and Analysis Center, California Institute of Technology, Pasadena, CA 91125, USA*

⁵*Department of Particle Physics and Astrophysics, Weizmann Institute of Science, Rehovot 76100, Israel*

Abstract. ZChecker is new, automated software for finding, measuring, and visualizing known comets in the Zwicky Transient Facility time-domain survey. ZChecker uses on-line ephemeris generation and survey metadata to identify images of targets of interest in the archive. Photometry of each target is measured, and the images processed with temporal filtering to highlight morphological variations in time. Example outputs show outbursts of comets 29P/Schwassmann-Wachmann 1 and 64P/Swift-Gehrels, and an asymmetric coma at C/2017 M4 (ATLAS).

1. Overview

Cometary science benefits from wide-field, time-domain optical surveys. Aside from the discovery of new comets, they can provide a better description of known objects through brightness variation with heliocentric distance and season; estimates of dust-to-gas ratio and its variation with time; and identification of cometary outbursts or other events for follow-up. The current challenge is to design tools that can facilitate rapid identification of anomalous behavior and enable follow-up studies of discovered events.

The Zwicky Transient Facility (ZTF) is an optical system using the Palomar Observatory 48-in Schmidt telescope (Bellm et al., in press). ZTF's time-domain surveys target a wide range of astrophysical phenomena (Graham et al., submitted). Solar System science is mainly piggybacked onto these surveys. A wide-field camera delivers a 47 deg² field of view, and 5 σ sensitivities near 20–21 mag in the *g*, *r*, and *i* filters. Images are processed and analyzed for transient sources and fed to an alert stream in near real time (Masci et al., in press; Patterson et al. 2019). All data are hosted at the Infrared Science Archive (IRSA).

The ZTF alert stream and ZTF Moving Object Discovery Engine (Masci et al., in press) regularly identify Solar System objects. However, these pipelines are optimized for point sources, and not guaranteed to find comets. As an alternative, we developed an ephemeris-based search tool, named ZChecker, custom designed for locating specific Solar System objects in the ZTF archive. Our primary goal is to promptly identify cometary outbursts for follow-up investigation. In the following sections we describe the search method and subsequent prompt cometary analyses.

2. Finding Solar System Objects

ZChecker is designed for daily searches for short ($\lesssim 1000$) lists of objects. The basis for the search engine is a spatial index via SQLite's R*Tree module (Guttman 1984; Beckmann et al. 1990). Here, the R-Tree's "rectangles" are four-dimensional bounding boxes: three spatial dimensions and one time. Each box is defined using five sky coordinates (four corners and the center) to account for the curvature of the celestial sphere. The polar singularities and the Right Ascension discontinuity at 0/24^h are avoided by using Cartesian coordinates. A separate R-Tree similarly indexes the object ephemerides. Searching with the indexes is accurate, but imprecise. However, the goal is to narrow down the number of potential matches from millions to tens, at which point slower, precise methods can be executed. R-Trees were previously discussed in the context of astronomy by Baruffolo (1999), and are used to find Solar System objects at IRSA and in the Keck Observatory Archive (Yau et al. 2011; Berriman et al. 2016).

Ephemerides are retrieved from online tools, either Minor Planet Center's Minor Planet and Comet Ephemeris System (MPES) or Jet Propulsion Laboratory's Horizons (Giorgini et al. 1997), using astroquery v0.3.9 (Ginsburg et al. 2018). Given a list of targets, ZChecker requests ephemerides for Palomar Observatory with adaptable time steps. Object lists are updated four times per year, but individual targets are added and updated as needed.

Once per day, the ZTF database at IRSA is queried for new science image metadata. The metadata are stored and indexed with the unique science product ID, and the image bounding boxes spatially indexed. Next, the ephemeris R-tree is queried for all tracks defined over the previous night. For each track, the observation R-tree is searched for overlapping boxes. For every match, a more precise target position is computed via spherical interpolation of the ephemeris, and compared to the image boundaries. If a match is found, a high-precision ephemeris and detailed observation geometry is retrieved from the MPES or Horizons, and a 5'×5' cutout is downloaded.

Daily searches typically take ~5 s, identifying up to ~100 observations of comets in ~10,000–30,000 data products. A single-object search for comets 2P/Encke and C/2017 M4 (PanSTARRS) in the full ZTF Partnership data archive (6 million images) takes 1 and 23 s identifying 17 and 478 observations, respectively. A full search for 528 comets nominally brighter than 25th mag yields 37,300 observations in 2380 s.

3. Photometry and Morphological Variations

The ZTF pipeline calibrates images using a filtered Pan-STARRS DR1 catalog (Masci et al., in press). Comets are extended objects with sizes that depend on, e.g., distance to the Sun and observer, or instrument sensitivity and bandpass. We centroid

on each comet, and then measure the total flux in a range of circular apertures. Automatic brightness plots using 10,000-km radius apertures are generated and inspected for anomalies. This fixed aperture size facilitates comet-to-comet comparisons.

In addition to photometric outburst discovery, we use temporal filtering to show morphological variations with time (cf. Schleicher & Farnham 2004). All images of a comet are projected into the ephemeris reference frame, with the projected Sun vector along the image $+x$ -axis. Images are photometrically scaled to a common heliocentric and geocentric distance, then median combined into nightly and two-week averages. The two-week average serves as the temporal reference for the nightly image. The difference or ratio two highlights morphological variations.

The default photometric scaling is designed for cometary comae: $r_h^{-4}\Delta^{-1}$, where r_h is heliocentric distance and Δ is observer-comet distance. This scaling does not account for dust phase darkening, and comets follow a wide range of heliocentric distance slopes, but in practice it works for most cases as a quick-look tool.

Based on ZChecker-produced data products, we independently discovered an outburst of comet 64P/Swift-Geherls (Kelley et al. 2018), and confirmed or rejected a few others. Moreover, we have identified a dust feature in the coma of comet C/2017 M4 (ATLAS). Example data are provided in Fig. 1.

Acknowledgments. M. Kelley acknowledges support from the NASA/University of Maryland/Minor Planet Center Augmentation through the NASA Planetary Data System Cooperative Agreement NNX16AB16A. Q. Ye acknowledges support from the GROWTH (Global Relay of Observatories Watching Transients Happen) project funded by the National Science Foundation PIRE program under Grant No 1545949.

Based on observations obtained with the Samuel Oschin 48-inch Telescope at the Palomar Observatory as part of the Zwicky Transient Facility project. Major funding has been provided by the U.S. National Science Foundation under Grant No. AST-1440341 and by the ZTF partner institutions: the California Institute of Technology, the Oskar Klein Centre, the Weizmann Institute of Science, the University of Maryland, the University of Washington, Deutsches Elektronen-Synchrotron, the University of Wisconsin-Milwaukee, and the TANGO Program of the University System of Taiwan.

This research made use of: the NASA/IPAC Infrared Science Archive, which is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration; Astropy, a community-developed core Python package for Astronomy (Astropy Collaboration et al. 2013); and, sbpy a community-driven Python package for small-body planetary astronomy supported by NASA PDART Grant No. 80NSSC18K0987.

References

- Astropy Collaboration, et al. 2013, *A&A*, 558, A33. 1307.6212
- Baruffolo, A. 1999, in *Astronomical Data Analysis Software and Systems VIII*, edited by D. M. Mehringer, R. L. Plante, & D. A. Roberts, vol. 172, 375
- Beckmann, N., Kriegel, H.-P., Schneider, R., & Seeger, B. 1990, in *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA: ACM), 322
- Bellm, E. C., et al. *PASP*. In press
- Berriman, G. B., et al. 2016, in *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913, 99130I
- Ginsburg, A., et al. 2018, *astropy/astroquery*. URL [dx.doi.org/10.5281/zenodo.1234036](https://doi.org/10.5281/zenodo.1234036)

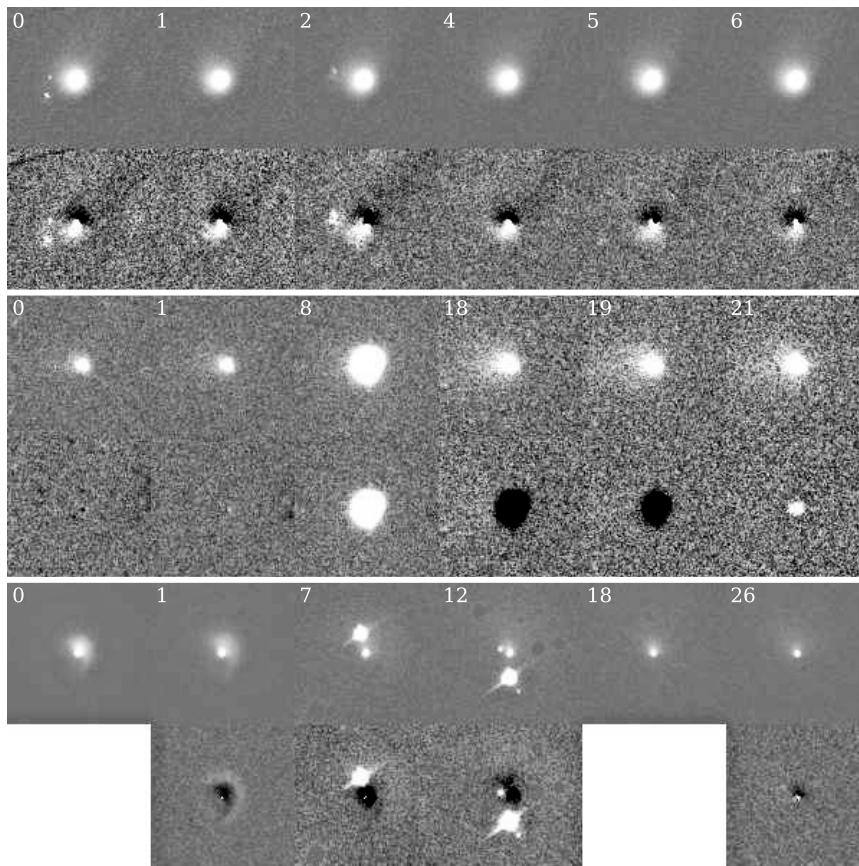


Figure 1. Time series and temporal filtered images of three comets. The time series is the first row and the reference subtracted data the second row. The sun direction is to the right. Images in each row are relatively scaled (linear scaling), and labeled with the relative time offset in days. (Top) C/2017 M4 (ATLAS), with residuals due to a strongly asymmetric coma and a rapid counter-clockwise rotation of the projected velocity vector. (Center) Outburst of 64P/Swift-Gehrels, the over-subtraction on days 18 and 19 caused by the outburst in the reference image. (Bottom) Outburst of 29P/Schwassmann-Wachmann 1 and propagating shell of dust.

- Giorgini, J. D., et al. 1997, in Bulletin of the American Astronomical Society, vol. 28, 1099
 Graham, M. J., et al. PASP. Submitted
 Guttman, A. 1984, in Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data (New York, NY, USA: ACM), 47
 Kelley, M. S. P., Bodewits, D., & Ye, Q. Z. 2018, Central Bureau Electronic Telegrams, 4544, 1
 Masci, F. J., et al. PASP. In press
 Patterson, M. T., et al. 2019, PASP, 131
 Schleicher, D. G., & Farnham, T. L. 2004, in Comets II, edited by Festou, M. C., Keller, H. U., & Weaver, H. A. (University of Arizona Press, Tucson), 449–469
 Yau, K. K., Groom, S., Teplitz, H., Cutri, R., & Mainzer, A. 2011, in American Astronomical Society Meeting Abstracts #217, vol. 217, 333.18

Session X

Time Domain Astronomy

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Data Challenges of the VO in Time Domain Astronomy

Ada Nebot

*CDS, Observatoire Astronomique de Strasbourg, Université de Strasbourg,
CNRS, UMR 7550, 11 rue de l'Université, 67000, Strasbourg, France*
ada.nebot@astro.unistra.fr

Abstract. Surveys specifically designed to monitor the transient sky have opened the window for discovery and exploration through Time Domain. Source classification and transmission of the alerts for further follow-up as well as analyzing possible periodicity in variable sources poses a challenge with the huge amounts of data synoptic missions are providing. We will review some of the challenges of Time Domain data and we will share some of the tools and services that exist or are being built within the Virtual Observatory to discover, access, visualize and analyze data adding the Time Domain aspect of data.

1. Introduction

Although Time Domain astronomy is currently a hot topic, it has been of great relevance for many decades. Great advances in our understanding of the Universe have been possible thanks to the analysis of variable sources. “The Great Debate of 1920 on the Scale of the Universe” was solved thanks to two major discoveries: the observed period-luminosity relation of Cepheid stars and the discovery of such stars in galaxies other than the Milky Way. The observed relation between the luminosity and the decay time of SNe Ia was the first direct evidence of the cosmic acceleration, as discussed during “The Great Debate of 1998 on the Nature of the Universe”. These two examples reflect how the study of variable sources, periodic or transient phenomena, can help us to answer fundamental questions in astronomy.

From Cepheid stars to SNe Ia explosions, nowadays we know that many different type of sources show variability over different time scales. In a very large sense Time Domain astronomy can be defined as *the study of variability of astronomical objects over different time-scales*. Variable phenomena can be classified into three different groups: 1) periodic phenomena such as binary orbits of stars or extra-solar planets, stellar rotation, stellar pulsation, ... ; 2) transient phenomena as seen in supernova explosions, gamma-ray bursts, nova, X-ray bursts, transits, gravitational micro-lensing, flares, tidal disruption events, ... ; 3) stochastic phenomena as in accretion in cataclysmic variables, X-ray binaries, ... The luminosity of these sources can vary over different time scales, from milliseconds to hundreds of years, covering more than five orders of magnitude. Even more, this variation can have different associated time-scales when observed at different wavelengths. Therefore source classification based on characteristic time-scale variations often needs a multi-wavelength approach, and sometimes even a multi-messenger approach is needed as demonstrated by the discovery of the electromagnetic counterpart to the GW170817 event.

Some of the challenges Time Domain astronomy faces include fast cross-matching of millions of sources covering large areas of the sky, source identification and classification, planning of observations, transmission of information, visualization and coordination tools, period search algorithms over big and heterogeneous datasets. These are all data-challenges and the Virtual Observatory (VO) should match the common needs of the different scientific use cases.

2. The IVOA

The International Virtual Observatory Alliance (IVOA) is the vision that astronomical datasets and other resources should work as a seamless whole. Many projects and data centers worldwide are working towards this goal. The International Virtual Observatory Alliance (IVOA) is an organization that debates and agrees the technical standards that are needed to make the VO possible. It also acts as a focus for VO aspirations, a framework for discussing and sharing VO ideas and technology, and body for promoting and publicizing the VO. The VO is integrated in many astronomy data centers and archives and is often behind the scenes. There are huge benefits from shared software components and the VO enables many scientific capabilities just not possible otherwise such as all sky astronomy. The VO is for research astronomers, data centers and archives, software developers, educators, etc. and the idea behind is to allow in a seamless way for the user to discover & access, visualize & analyses data through services & tools.

Within the IVOA there are six working groups and seven interest groups, among which there is a Time Domain interest group¹. There are two interoperability meetings per year and specific email lists associated to each working or interest group for discussion of topics of interest and these are all completely open to participation.

3. Time Domain astronomy challenges and status of the IVOA

Time Domain astronomy is currently a scientific priority of the IVOA. In this section some data challenges related to enabling Time Domain astronomy in the VO and their status are highlighted:

3.1. Multi-wavelength/messenger approach is (sometimes) needed

To combine data from different missions covering different wavelengths fast and efficient cross-matching techniques are developed within the framework of the IVOA (see e.g. <http://cdsxmatch.u-strasbg.fr/>). Cross-matching tools take into account mostly only positional information. While some cross-matching tools have been developed to take into account the multi-wavelength aspect of sources (Motch et al. 2017), the temporal dimension has not yet been fully integrated. For fast moving objects the location in the sky at two different moments varies and a simple positional cross-match could lead to erroneous results. Information on the time of observation and on the proper motion is needed in order to perform a proper cross-match. A possible enhancement for cross-matching capabilities would be to add the temporal dimension as extra information.

¹<https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaVOEvent>

Source characterization and classification on the basis of their variability in a multi-wavelength space and of large and heterogeneous data using machine learning or deep learning techniques are questions that are being discussed within the IVOA Knowledge Discovery Interest Group².

3.2. Alerts and follow-up observations

Transient events can trigger alerts for which follow-up observations are crucial for identification of the source. While some transient events fade out quickly in a specific wavelength, they might still be visible at other wavelengths. Generation and transmission of such alerts between telescopes in an automated way can be done through the IVOA standard protocols VOEvent (Seaman et al. 2011) and its Transmission Protocol (Swinbank et al. 2017). These two protocols are nowadays widely used by the community, from Fast Radio Bursts (Petroff et al. 2017) to Solar and Planetary Sciences (Cecconi et al. 2018).

3.3. Navigation through the data

Many different Applications and VO-compliant tools allow to navigate through images and catalogues in space (Aladin, firefly, SkyView, DS9,...). These tools are evolving towards a better data interoperability by integrating the temporal dimension to enable the display of measurements as a function of time, while simultaneously visualizing the single-epoch images. One of the challenges of the IVOA is to be able to search databases by a time constraint. While the IVOA standard for searching data based on location on the sky exists for more than a decade through the cone search protocol (Plante et al. 2008), the equivalent for time, i.e. a simple query protocol for retrieving records from a catalog of astronomical sources based on an interval of time, is currently not existing. Definition of the minimum metadata to describe the time and a standard annotation would allow data providers to enable interoperability. In this context, a new element called TIMESYS has been proposed by the IVOA. This element describes the minimum metadata needed to annotate data containing time³. This element, TIMESYS, has three fundamental elements: the time scale (TT, TAI, UTC,...), the reference position (TOPOCENTER, BARYCENTER, ...), and the offset which might have been subtracted to the data (it is very common to subtract a certain date to time values for keeping sufficient accuracy). This new element is currently being discussed and if arrived to consensus it will be integrated into the VO format for tables VOTable (Ochsenbein et al. 2013).

Hierarchical Progressive Survey (HiPS Fernique et al. 2017) and Multi-Order Coverage map (MOC, Fernique et al. 2014), both IVOA specifications widely used by the community, allow the users to quickly navigate through datasets in space (collection of images or catalogues). HiPS is a hierarchical scheme for the description, storage and access of sky survey data, HiPS allows the possibility of creating HiPS-cubes where the third dimension typically used is the wavelength, but nothing prevents users to create HiPS-cubes where the third dimension is a temporal to create a time-view of the sky. Based on a hierarchical approach of the sky a MOC gives the approximate area covered by different missions, enabling the capacity of comparing coverage maps of

²<https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaKDD>

³<http://ivoa.net/documents/Notes/TimeSys/index.html>

large datasets in a seamless way. Users can make fast operations such as intersections and unions of several missions. Based on the same technology as MOC, time-MOCs, T-MOCs take into account the time dimension of data, allowing users to see if two missions overlapped in time in a very fast way (see poster contribution P8 from Fernique et al.). In the top panel of Fig. 1 a T-MOC shows the coverage of a certain dataset in time, together with the visualization of the data in the bottom panel. Ideally space and time will be combined to create a unique coverage map.

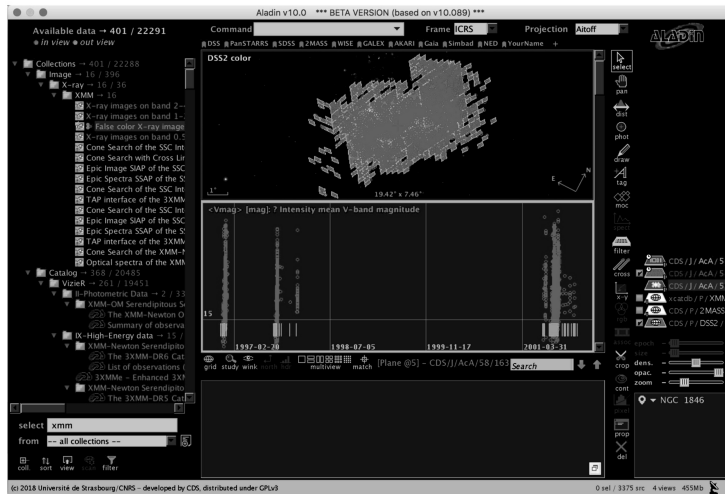


Figure 1. Aladin showing a) upper panel: positions of sources in the sky (red open circles) on top of an X-ray image and the spatial coverage (green semi transparent tiles); b) lower panel: the variation of the magnitude of the same sources over time (red open circles) as well as their coverage in time (green vertical lines).

3.4. Visualization of light-curves

A quick look at the spectral energy distribution of a source can help to understand the nature of the object of study. With this idea in mind, photometric viewers are now widely used. Based on the location on the sky and a search radius around that position, photometric viewers display fluxes or magnitudes against frequency (or wavelength) of catalogue data including values in well defined filters (see e.g. <http://vizier.u-strasbg.fr/vizier/sed/>). For that purpose a huge effort had to be carried out to characterize filters and collect that information in a standardized way (see the filter profile system of the SVO⁴). A quick look at the light-curve of an object can also help to say something about its nature. Ideally photometric viewers would have the capability to include the time as a possible extra dimension, that is a *variability photometric viewer*, to help the user have a quick view of the photometric variability of the source (see https://wiki.ivoa.net/internal/IVOA/InterOpOct2017TDIG/VizieR_

⁴<http://svo2.cab.inta-csic.es/svo/theory/fps/>

timeSeries.pdf). The key for being able to create such a tool or service is to have a standard annotation for time (e.g. TIMESYS) and to transform into a pivot format in which the data can be compared. Within the VO framework we are working towards such an implementation.

3.5. Analysis of variance of phenomena

To analyze periodic variability there are different algorithms, parametric (Fourier based) and non-parametric. The tool Period04 (Lenz & Breger 2004)⁵, interoperable through SAMP Taylor et al. (2012), implements different parametric methods to analyze variability in data and find any possible periodicity. If one or more periods are found the data can be folded to visualize the data in phase space. This tool is widely used and a new version will soon be released (priv. communication). Combining Time Series for a variability analysis is a tricky thing, since periodicity might depend on the quantity that is measured (magnitudes in different filters, radial velocities) and since the user has to be sure that the times values are all brought to the same time frame. Combining a time value at the Earth topocenter with a time value at the Solar System barycenter could lead to significant errors in an Analysis of Variance, since the time values can differ for more than 8 minutes between these two reference position. The usage of an standard annotation of time values could help in preventing such mistakes and VO-compliant tools performing periodicity analysis could also benefit from an element such as the proposed TIMESYS.

3.6. Coordination & transmission of information

Both Time Domain and Multi-messenger astronomy have a common need for coordination of observations in multi-national collaborations, ranging from planning of observations, collection of available information, to the transmission of information. Thanks to the GR170817 gravitational wave detection and the enormous follow-up campaign which involved more than 70 telescopes ground- and space-based lead to associating the event to colliding neutron stars (Abbott et al. 2017). This discovery marked the first cosmic event observed in both gravitational waves and light. As a side effect of the process leading to such discovery, community awareness on the need of good practice on coordination, documentation and communication has increased. Planning observations involves knowledge on target visibility, telescope time availability, and prioritization of the scheduled time. Different visibility services (e. g. <http://catserver.ing.iac.es/staralt/>) and schedule planning (e. g. http://archive.eso.org/wdb/wdb/eso/sched_rep_arc/form) exist but they have inputs and outputs which are very heterogeneous. Some level of standardization would facilitate the users' tasks to plan and organize observations. With that aim two protocols are being developed within the IVOA framework: an Object Visibility Simple Access Protocol ⁶ and an Observation Locator Table Access Protocol ⁷. A facility database is also being built with the aim of documenting observatories, locations, telescopes, instruments, filters, etc. that are (or have been) available to the community (Perret et al.

⁵<https://www.univie.ac.at/tops/Period04/>

⁶<http://ivoa.net/documents/ObjVisSAP/index.html>

⁷<http://ivoa.net/documents/ObsLocTAP/index.html>

2018). In this context, one of the ideas that is also being discussed and prototyped is the possibility of having an exchange platform dedicated to such type of follow-up campaigns (see e. g. <http://multi-messenger.asterics2020.eu/>).

4. Summary and conclusions

Time Domain astronomy is currently a scientific priority for the IVOA. Different working groups, in particular Applications, Data Model and Data Access Working Groups are working together with the Time Domain Interest Group to make discovery, access, visualization and analysis of data based on time possible, through the definition of standard protocols that could be used by the community. Details on all available IVOA standard protocols can be found under the Document section of the IVOA webpage⁸ and we would like to encourage people to join this international effort for further development and improvement.

References

- Abbott, B. P., Abbott, R., Abbott, T. D., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., Adhikari, R. X., Adya, V. B., & et al. 2017, *ApJ*, 851, L16. 1710.09320
- Cecconi, B., Le Sidaner, P., Tomasik, L., Marmo, C., Garnung, M. B., Vaubaillon, J., André, N., & Gangloff, M. 2018, arXiv e-prints. 1811.12680
- Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017, HiPS - Hierarchical Progressive Survey Version 1.0, IVOA Recommendation 19 May 2017. 1708.09704
- Fernique, P., Boch, T., Donaldson, T., Durand, D., O'Mullane, W., Reinecke, M., & Taylor, M. 2014, MOC - HEALPix Multi-Order Coverage map Version 1.0, IVOA Recommendation 02 June 2014. 1505.02937
- Lenz, P., & Breger, M. 2004, in *The A-Star Puzzle*, edited by J. Zverko, J. Ziznovsky, S. J. Adelman, & W. W. Weiss, vol. 224 of IAU Symposium, 786
- Motch, C., Carrera, F., Genova, F., Jiménez-Esteban, F., López, M., Michel, L., Mingo, B., Mints, A., Nebot, A., Pineau, F.-X., Rosen, S., Sanchez, E., Schwöpe, A., Solano, E., & Watson, M. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of Astronomical Society of the Pacific Conference Series, 165. 1609.00809
- Ochsenbein, F., Taylor, M., Williams, R., Davenhall, C., Demleitner, M., Durand, D., Fernique, P., Giaretta, D., Hanisch, R., McGlynn, T., Szalay, A., & Wicenec, A. 2013, VOTable Format Definition Version 1.3, IVOA Recommendation 20 September 2013
- Perret, E., Louys, M., Buga, M., & Lesteven, S. 2018, in *European Physical Journal Web of Conferences*, vol. 186 of European Physical Journal Web of Conferences, 04002
- Petroff, E., Houben, L., Bannister, K., Burke-Spolaor, S., Cordes, J., Falcke, H., van Haren, R., Karastergiou, A., Kramer, M., Law, C., van Leeuwen, J., Lorimer, D., Martinez-Rubi, O., Rachen, J., Spitler, L., & Weltman, A. 2017, arXiv e-prints. 1710.08155
- Plante, R., Williams, R., Hanisch, R., & Szalay, A. 2008, Simple Cone Search Version 1.03, IVOA Recommendation 22 February 2008. 1110.0498
- Seaman, R., Williams, R., Allan, A., Barthelmy, S., Bloom, J., Brewer, J., Denny, R., Fitzpatrick, M., Graham, M., Gray, N., Hessman, F., Marka, S., Rots, A., Vestrand, T., & Wozniak, P. 2011, Sky Event Reporting Metadata Version 2.0, IVOA Recommendation 11 July 2011. 1110.0523

⁸<http://ivoa.net/documents/>

- Swinbank, J. D., Allan, A., & Denny, R. B. 2017, VOEvent Transport Protocol Version 2.0, IVOA Recommendation 20 March 2017
- Taylor, M., Boch, T., Fitzpatrick, M., Allan, A., Fay, J., Paiono, L., Taylor, J., & Tody, D. 2012, Simple Application Messaging Protocol Version 1.3, IVOA Recommendation 11 April 2012



(from left to right) Pascal Ballester and Mauricio Solar working on the 2017 ADASS proceedings. (Photo: Peter Teuben)



Banquet (Photo: Peter Teuben)



Roberto Pizzo, Yan Grange and Lisa Press preparing the ADASS2019 booth. (Photo: Elizabeth Warner)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

The ZTF Alert Stream: Lessons from the First Six Months of Operating an LSST Precursor

Mario Juric¹, Eric C. Bellm¹, Maria T. Patterson¹, V. Zach Golkhou¹, and Benjamin Rusholme²

¹ *DIRAC Institute, Department of Astronomy, University of Washington, 3910 15th Avenue NE, Seattle, WA 98195, USA; mjuric@astro.washington.edu*

² *IPAC, California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125, USA;*

Abstract. The Zwicky Transient Facility (ZTF; Bellm et al. 2019) is an optical time-domain survey that is currently generating about one million alerts each night for transient, variable, and moving objects. The ZTF Alert Distribution System (ZADS; Patterson et al. 2019) packages these alerts, distributes them to the ZTF Partnership members and community brokers, and allows for filtering of the alerts to objects of interest, all in near-real time. This system builds on industry-standard real-time stream processing tools: the Apache Avro binary serialization format and the Apache Kafka distributed streaming platform. ZADS routinely transports 0.6 to 1.2 million alerts per night (amounting to 70GB/night), and has handled peaks of over 2 million alerts/night with no technical issues.

1. Introduction

The ZTF Alert Distribution System (Patterson et al. 2019) is a near-real-time streaming platform for fast distribution and filtering of sources detected by the ZTF's image processing pipelines (Masci et al. 2019). ZADS' role is to ingest alerts from the output of image differencing object detection processing, and deliver them to downstream brokers and science users (Figure 1). ZADS aims to make science quality alerts from ZTF available to downstream users within 20 minutes of observation. The ZADS system is a gateway to enabling real-time astronomical time domain science with ZTF. This includes the discovery of young supernovae, the identification of stellar variables, and the search for electromagnetic counterparts to gravitational wave sources.

At present, this system primarily serves four major public event broker systems which further distribute and provide access to the broader community: the Arizona NOAO Temporal Analysis and Response to Events System (ANTARES; Narayan et al. 2018); Lasair (<http://lasair.roe.ac.uk/>), a broker serving the U.K. community; an initiative based in Chile called Automatic Learning for the Rapid Classification of Events (ALeRCE; <http://alerce.science/>); and Las Cumbres Observatory's Make Alerts Really Simple (MARS; <https://mars.lco.global/>).

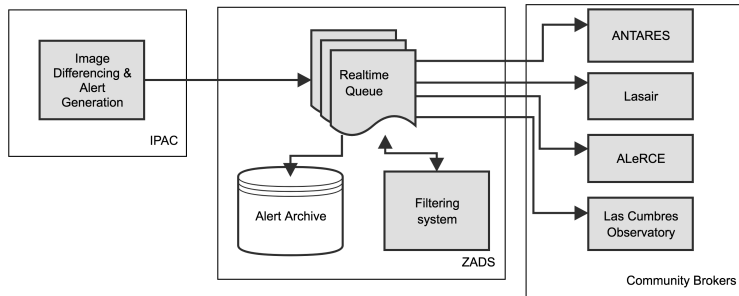


Figure 1. Schematic overview of ZTF alert flow (reproduced from Patterson et al. 2019).

2. Alert formatting: Apache Avro

For alert serialization, ZADS uses Apache Avro¹, a structured binary format with JSON-based schemas common in the Hadoop ecosystem. Serializing to Avro allows for smaller alert packet sizes, with an average reduction of a factor of six relative to XML packaging of the same data. A secondary benefit is in speed of serialization and deserialization with observed speedups of up to 40 times (Maeda 2012). For ZTF, we find that Avro alerts are approximately half the size of the same alert content serialized to XML with the schema definition embedded, and one-third the size without schema. Compressed, Avro alerts are approximately 35% smaller than compressed XML.

Avro packets can be received and parsed with readily available client libraries for nearly all major programming languages, including Python. We do note that these libraries show a significant variance in quality, particularly regarding the trade-off between speed and functionality. At the moment, the *fastavro*² library appears to provide a reasonable balance between the two.

The current ZADS Avro message format and data model is ZTF-specific, and informed by the needs of ZTF science. As an example, the detection history of each alert is structured so as to allow for simple extraction and processing of lightcurves. Secondly, ZTF alerts include small (FITS-formatted) image cutouts of the detections, as opposed to the URI to a location where they can be accessed. By comparison, VO-Event is designed to be more general, but with some loss of efficiency and end-user ease of use. The lessons learned from using the ZTF alert stream could provide useful input for evolving VOEvent in anticipation of alert streams from massive surveys such as the LSST.

¹<http://avro.apache.org/>

²<https://github.com/fastavro/fastavro>

3. Alert distribution: Apache Kafka

For alert transport ZADS employs Apache Kafka (Kreps et al. 2011), an open source stream processing software platform. Kafka is especially suitable for applications requiring high throughput and low latency message transport, and has been deployed for industry applications transmitting over 10^{12} messages per day. Kafka employs a distributed architecture, allowing one to deploy a fault-tolerant cluster of Kafka “brokers”. For ZADS, we have deployed Kafka in a number of configurations, including single-node installs, as well as a three-broker cluster managed with Docker.

Kafka implements a classical pub/sub pattern. Messages are injected into message queues (called “topics”) by “producers”. Within each topic, messages are written into “partitions”. The topics can be thought of as a named stream which can be subscribed to or a single log that can be sequentially read from. The messages are ordered within the partitions, though not within the topic itself. Downstream “consumers” subscribe to individual topics, potentially reading in parallel from different partitions.

Kafka provides “at-least-once” guarantees for message delivery. The system tracks the latest message offset acknowledged to have been read by a consumer from each topic-partition. Should a consumer disconnect, it will re-read the stored offset upon reconnection. This both ensures that those listening to the stream will not lose messages in exceptional situations, but also allows consumers to rewind to past offsets in order to reprocess data. This rewind feature is an enhancement over protocols typically used to transport VOEvents, such as the VO Event Transport Protocol (VTP; Allan et al. 2017)

3.1. Alert filtering plans

At the source, the ZTF alert stream is unfiltered – it includes all scientifically usable detections collected by ZTF. To make it friendlier to the end-users, pre-defined stream filters for a number of science use cases are in the process of being tested. These will be securely and redundantly deployed as Python code, running in Docker containers. Architecturally, each filter is both a Kafka consumer and producer, reading from the main topic and writing into a separate, filtered, topic. After they’re read, alert messages are deserialized into Python dictionaries and passed to a Python function that returns True if the alert should be retained, and False if not. Alerts that pass the filter are serialized and written to a filtered topic.

The above architecture is extendable to custom, user-supplied, filters. Our long-term goal with ZADS is to make it easy for users to create such filters, and be forwarded only the alert messages of interest to their particular science case. Practically, we wish to enable the reduction in volume from $O(1M)$ to $O(10)$ candidates that the end-user receives, making human inspection and rapid follow-up a reasonable possibility.

4. Performance and metrics

ZTF typically generates between 600,000 to 1.2 million alerts per night (depending on the season). An individual, Avro-serialized, alert is typically about 60 KB in size, dominated by the included FITS cutouts. The nightly volume of ZTF alerts can therefore amount to over 70 GB of data. ZADS routinely scaled to these volumes, and has achieved throughput of over 2 million alerts with no technical issues.

The acquisition-to-scientist latency of ZTF’s complete real-time processing pipeline is roughly 20 minutes, dominated by the data reduction time. The alert packaging and

transmission element is significantly shorter than that, at about 6 seconds. This is consistent with simulations prior to the start of the survey – the serialization of 1,000 ZTF alerts into Avro, writing to a Kafka topic, and transfer of those data to a consumer was measured at 4.2 seconds. From the main Kafka hub at the University of Washington to a cloud-based Kafka instance, data transfer rates of about 100 MBps have been observed. This equates to over 80k alerts/minute or 6.6k alerts in 5 seconds, tantalizingly close to LSST goals (10k alerts in 5 seconds).

5. Summary

The ZADS platform is the first successful demonstration of $O(10^6)$ message scale alert distribution in an astronomical context, using industry standard tools and serialization formats. It has successfully transmitted more alerts than have ever been distributed before, on the order of one million per night. Given its success on ZTF, we're hopeful that lessons learned from ZADS will provide guidance for the evolution of astronomical practice and standards for event distribution, in preparation for the Large Synoptic Survey Telescope (Ivezic et al. 2008).

Acknowledgments. Based on observations obtained with the Samuel Oschin Telescope 48-inch and the 60-inch Telescope at the Palomar Observatory as part of the Zwicky Transient Facility project, a scientific collaboration among the California Institute of Technology, the Oskar Klein Centre, the Weizmann Institute of Science, the University of Maryland, the University of Washington, Deutsches Elektronen-Synchrotron, the University of Wisconsin-Milwaukee, and the TANGO Program of the University System of Taiwan. Further support is provided by the U.S. National Science Foundation under Grant No. AST-1440341.

M. Juric acknowledges the support of the Washington Research Foundation Data Science Term Chair fund, and the UW Provost's Initiative in Data-Intensive Discovery.

E. Bellm is supported in part by the NSF AAG grant 1812779 and grant #2018-0908 from the Heising-Simons Foundation.

M. Patterson, E. Bellm, and M. Juric acknowledge support from the University of Washington College of Arts and Sciences, Department of Astronomy, and the DIRAC Institute. University of Washington's DIRAC Institute is supported through generous gifts from the Charles and Lisa Simonyi Fund for Arts and Sciences, and the Washington Research Foundation.

References

- Allan, A., Denny, R. B., & Swinbank, J. D. 2017, ArXiv e-prints. 1709.01264
- Bellm, E. C., Kulkarni, S. R., Graham, et al. 2019, PASP, 131, 018002
- Ivezic, Z., Tyson, J. A., et al. 2008, ArXiv e-prints. 0805.2366
- Kreps, J., Narkhede, N., Rao, J., et al. 2011, in Proceedings of the NetDB, 1
- Maeda, K. 2012, in Digital Information and Communication Technology and its Applications (DICTAP), 2012 Second International Conference on, 177
- Masci, F. J., Laher, R. R., Rusholme, et al. 2019, PASP, 131, 018003
- Narayan, G., et al. 2018, ApJS, 236, 9. 1801.07323
- Patterson, M. T., Bellm, E. C., Rusholme, B., Masci, F. J., Juric, M., Krughoff, K. S., Golkhou, V. Z., Graham, M. J., Kulkarni, S. R., Helou, G., & Zwicky Transient Facility Collaboration 2019, PASP, 131, 018001

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

A GPU Implementation of the Harmonic Sum Algorithm

Karel Adámek and Wesley Armour

Oxford e-Research Centre, Department of Engineering Science, University of Oxford, Oxford, OX1 3QG, UK; karel.adamek@oerc.ox.ac.uk

Abstract. Time-domain radio astronomy utilizes a harmonic sum algorithm as part of the Fourier domain periodicity search, this type of search is used to discover single pulsars. The harmonic sum algorithm is also used as part of the Fourier domain acceleration search which aims to discover pulsars that are locked in orbit around another pulsar or compact object. However porting the harmonic sum to many-core architectures like GPUs is not a straightforward task. The main problem that must be overcome is the very unfavorable memory access pattern, which gets worse as the dimensionality of the harmonic sum increases. We present a set of algorithms for calculating the harmonic sum that are more suited to many-core architectures such as GPUs. We present an evaluation of the sensitivity of these different approaches, and their performance. This work forms part of the AstroAccelerate project (Armour et al. 2013) which is a GPU accelerated software package for processing time-domain radio astronomy data.

1. Introduction

Detecting pulsars in time-domain radio astronomy using Fourier transform based techniques is a convenient and computationally efficient way to extract the faint periodic pulses from the noise in which they sit. However this technique, called periodicity searching, has some pitfalls. One of these is that the power contained in pulsar signal is spread into multiple harmonics in the calculated power spectra. The incoherent harmonic sum algorithm is one way to rectify this. In pulsar searches the observed time-series are first de-dispersed and then transformed using an FFT into frequency space. Then the harmonic sum is applied, which aims to sum the signal present and average out the noise. This is done by calculating partial sums of an increasing number of harmonics.

The harmonic sum algorithm sums the power that is spread across multiple harmonics back into a single Fourier bin. This increases the signal-to-noise ratio of detected pulsars and allows us to detect weaker pulsars as a result. The two-dimensional harmonic sum is also the next step after the Fourier domain acceleration search technique (FDAS) (Ransom et al. 2002), which searches for accelerated pulsars. There is a GPU implementation of FDAS by Dimoudi et al. (2018), there is also a GPU implementation of a two-dimensional harmonic sum for PRESTO by Luo (2013).

The harmonic sum is given by equation

$$h(n)_H = \frac{1}{\sqrt{H}} \sum_{i=1}^H x\left(\frac{ni}{H}\right), \quad (1)$$

where H is the number of harmonics summed. The number of harmonics we need to sum is governed by the duty-cycle of the pulsar we are looking for.

The main problem with summing harmonics is that the peaks of the pulsar signal which we aim to add together can be, for higher harmonics, shifted by the number of frequency bins equal to the currently summed harmonic. The optimal harmonic sum performs all possible sums, which results in the best signal-to-noise ratio. However, the shift in position of the pulsar signal for higher harmonics creates an unfavorable memory access pattern and also load balancing issues.

The harmonic sum algorithm is a standard in many software packages which process radio astronomy data, such as SIGPROC (Lorimer 1999) or PRESTO (Ransom 2002). However none of these packages have the harmonic sum in a computationally accelerated form.

2. Algorithms

We have investigated a set of different harmonic sum algorithms, each algorithm has different properties and so can be used for different purposes. Some algorithms have good sensitivity, but suffer in performance and visa versa. We have implemented these harmonic sum algorithms on both the CPU and the GPU. The CPU versions of the harmonic sum are naively parallelized across the number of time-series processed. In the case of our GPU implementations the parallelization strategy depends on the harmonic sum algorithm under consideration.

We have compared our algorithms to the widely accepted harmonic sum based equation 1, which is widely used, for example in the SIGPROC software package.

In this paper we introduce several harmonic sum algorithms which trade sensitivity for performance. The most obvious way to increase performance is to limit the number of sums examined by the algorithm. The algorithm Max HRMS works by looking for a maximum at possible locations of the peak and adds this value to the partial sum for that harmonic sum.

$$h(n)_H = \frac{1}{\sqrt{H}} \sum_{i=1}^H \max_{1 \leq j \leq i} (x(in + j)) . \quad (2)$$

In freq. bin HRMS we only add values for bins which are integer multiples of the fundamental frequency, that is

$$h(n)_H = \frac{1}{\sqrt{H}} \sum_{i=1}^H x(in) . \quad (3)$$

Lastly, in Greedy HRMS we recursively add the value in the appropriate bin or one of its neighbors depending what is bigger to the partial sum.

$$h(n)_{H+1} = h(n)_H + \max (x(Hn + j), x(Hn + j + 1)) , , \quad (4)$$

where j is increased by one if $x(n+j+1)$ is selected.

3. Results

To measure the sensitivity of our algorithms we generated a time-series containing an artificial pulsar for which we have used the modified von Mises distribution. The same

approach was used by Ransom et al. (2002). White noise was added using a pseudo random number generator with a normal distribution. The time-series contains 20 seconds of observing data with a sampling time $t_s = 64\mu s$. We measure sensitivity using the signal-to-noise (SNR) ratio recovered by the algorithm for a pulsar of a given initial SNR in comparison to a standard algorithm based on equation 1.

The pulsar's frequency could fall in between frequencies of the Fourier bins which result from the discrete Fourier transformation. This occurs when peaks of higher harmonics are non-integer multiples of the fundamental frequency. This results in a lower recovered SNR. The recovered SNR by different algorithms for such cases are shown in Figure 1. We see that the highest SNR detected is at the frequency of the Fourier bins. There is the steep decline in recovered SNR for other pulsar periods. The frequency bin HRMS algorithm has for non-Fourier bin frequencies a sensitivity loss of more than 50%. This is expected since the algorithm does not take into account the shift of the peak for higher harmonics. The Max HRMS algorithm has lower recovered SNR values, but they are consistent throughout the Fourier bin.

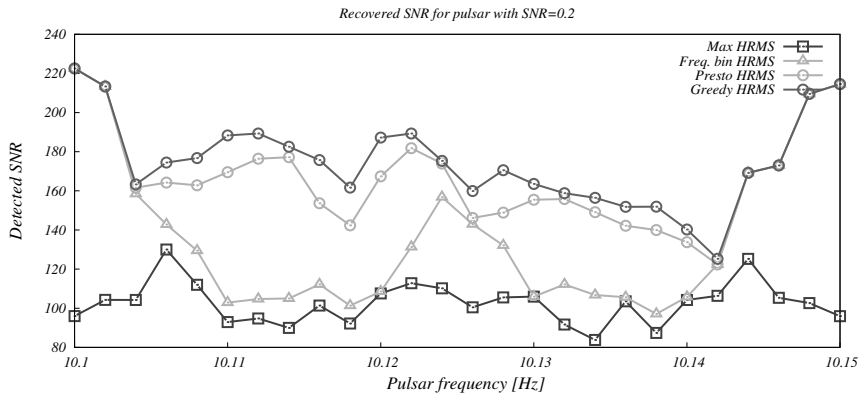


Figure 1. SNR recovered by different algorithms for pulsar frequencies which lie between discretised frequencies of the Fourier bins. Width of the bin is 0.05Hz.

The averaged recovered SNR for a wide range of pulsar frequencies is shown in Figure 2. All presented algorithms recover consistent values of averaged SNR regardless of pulsar frequency. We see that standard HRMS and greedy HRMS has similar averaged SNR. The freq.bin HRMS algorithm has an averaged SNR loss of about 20% and max HRMS algorithm of about 40%. Lastly, we present the performance of our implementations of the algorithms we have described here. For each algorithm we have implemented both CPU and GPU versions. The comparison in performance of the GPU versions to their respective CPU versions are shown in Table 1. This table also shows the speed-up of all GPU versions of these algorithms with respect to the harmonic sum algorithm based on equation 1 which we have used as our testing standard.

4. Conclusions

We have presented the sensitivities of several algorithms for the calculation of the harmonic sum. We have implemented a set of these algorithms on both CPU and GPU

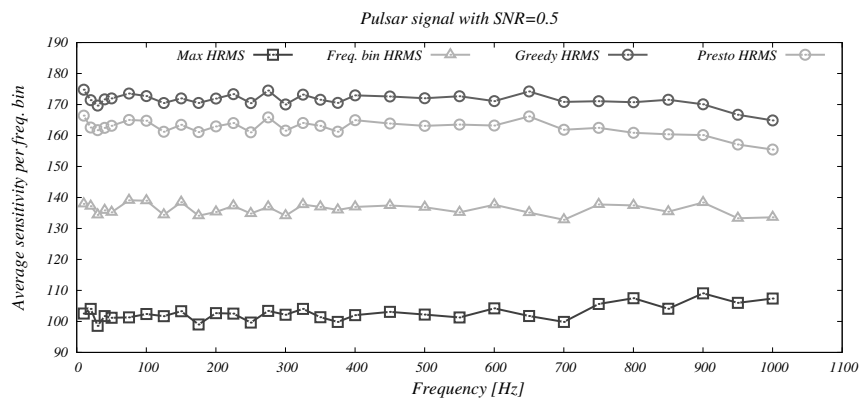


Figure 2. Averaged recovered SNR by different algorithms for a wide range of pulsar frequencies. All algorithms perform consistently on selected frequencies.

Table 1. Speed-up of GPU harmonic sum algorithms to their CPU counterparts and to our GPU implementation of the standard HRMS algorithm.

Algorithm	vs CPU	vs GPU Standard HRMS
Standard HRMS	250×	1.0×
Greedy HRMS	704×	6.4×
Freq. bin HRMS	110×	10.1×
Max HRMS	177×	3.9×

hardware and compared the performance of the CPU vs. GPU implementation for each algorithm. We have also compared GPU implementations of other algorithms against the performance of the standard HRMS algorithm. Our fastest algorithm is the freq. bin HRMS which has a sensitivity loss of approximately 20% of the signal. A good ratio of sensitivity and performance is given by the greedy HRMS algorithm which has high sensitivity and also high performance, further work to explore this is underway.

References

Armour, W., et al. 2013, Astroaccelerate, <https://github.com/AstroAccelerateOrg>
Dimoudi, S., et al. 2018, ArXiv e-prints. 1804.05335
Lorimer, D. 1999, Sigproc, <https://sourceforge.net/projects/sigproc>
Luo, J. 2013, Presto gpu, https://github.com/jintaoluo/presto_on_gpu
Ransom, S. M. 2002, Presto, <https://github.com/scottransom/presto>
Ransom, S. M., Eikenberry, S. S., & Middleditch, J. 2002, The Astronomical Journal, 124, 1788. astro-ph/0204349

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Prototype Implementation of a Web-Based Gravitational Wave Signal Analyzer: SNEGRAF

Satoshi Eguchi, Shota Shibagaki, Kazuhiro Hayama, and Kei Kotake

Fukuoka University, 8-19-1, Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan;
satoshiieguchi@fukuoka-u.ac.jp

Abstract. A direct detection of gravitational waves is one of the most exciting frontiers for modern astronomy and astrophysics. Gravitational wave signals combined with classical electro-magnetic observations, known as multi-messenger astronomy, promise newer and deeper insights about the cosmic evolution of astrophysical objects such as neutron stars and black holes. To this end, we have been developing an original data processing pipeline for KAGRA, a Japanese gravitational wave telescope, for optimal detections of supernova events. As a part of our project, we released a web application named SuperNova Event Gravitational-wave-display in Fukuoka (SNEGRAF) in autumn 2018. SNEGRAF accepts the users' theoretical waveforms more than $\sim 10^5$ data points directly with JavaScript, although the number can be typical for a supernova hunt by assuming a typical duration of the event and sampling rate of the detectors; a combination of recursive decimations of the original in the server-side program and an appropriate selection of them depending on the time duration requested by the user in a web browser achieves an acceptable latency. In this paper, we present the current design, implementation and optimization algorithms of SNEGRAF, and its future perspectives.

1. Introduction

In the framework of general relativity, a mass curves the space-time around it, and the curvature is observed as gravity. An accelerated motion of a mass generates a disturbance of space-time, which propagates in a vacuum in the form of waves; these waves are referred to as “gravitational waves.” Gravitational waves can penetrate even a very dense material, and carry the information of the space-time around a massive but compact astronomical object such as a neutron star and black hole. The first direct detection of a gravitational wave is known as GW150914, where a merger of two stellar-mass black holes took place (Abbott et al. 2016).

Multi-messenger astronomy, which utilizes observations of gravitational waves and neutrinos combined with those in multiple wavelengths, attracts a lot of attention recently since it promises a deeper understanding of the innermost part of a high energy astrophysical phenomenon. For example, roughly two explosion mechanisms are proposed for a core-collapse supernova to date, leading to outstanding differences in their gravitational waveforms (see Kotake 2013, and references therein). Hence, at Fukuoka University, we assembled a team to promote multi-messenger astronomy focusing on the physics of supernovae in April 2018. Goals of our mission are:

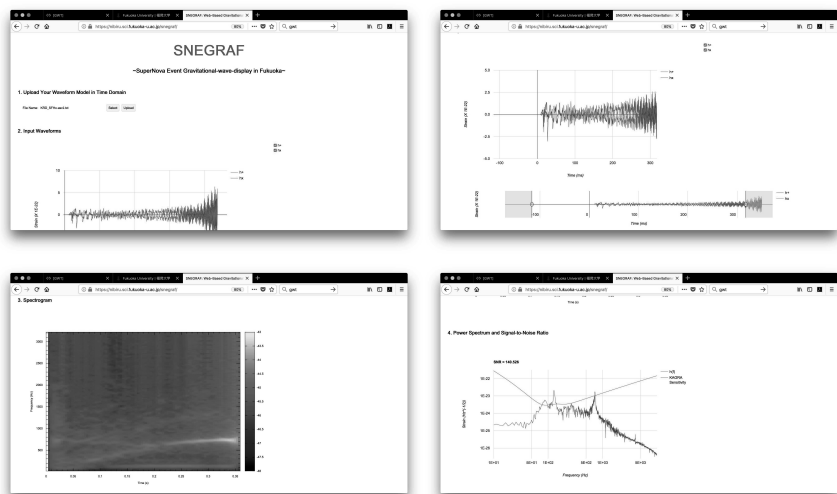


Figure 1. Screenshots of SNEGRAF. From left to right and top to bottom, the banner, waveform viewer, spectrogram, and power spectrum, respectively.

- Developing an original data processing pipeline for KAGRA, a Japanese gravitational wave telescope, to detect a supernova event at optimal efficiency,
- Providing data visualization and analysis software for the KAGRA observations to the world.

2. SNEGRAF

As the first step of our software releases, we have just made a web application named “SuperNova Event Gravitational-wave-display in Fukuoka (SNEGRAF; Fig. 1)” public in October 2018. SNEGRAF accepts a time series of h_+ and h_\times (two individual modes of a gravitational wave) in a character-separated-value (CSV) format as an input (Table 1), and displays the input waveforms, a corresponding spectrogram, and power spectrum together with the signal-to-noise ratio of the input signal and the analytic KAGRA sensitivity curve. The access url to SNEGRAF is <https://nibiru.sci.fukuoka-u.ac.jp/snegrarf/>.

Table 1. Details of an input file format for SNEGRAF. A pipe (|), comma (,), tab (\t), and white space are acceptable for a column separator. A hash (#) is regarded as a beginning of comments.

Column Number	1st	2nd	3rd
Content	Time (sec)	h_+	h_\times

SNEGRAF is a simple Ajax application hosted on a Java servlet. Since we have quite limited human resources and utilize existing software libraries written in either

C/C++ or Java, we adopt GWT¹, which generates both server-side and client-side codes from a single Java source file, for an application framework. Google Charts² and its GWT binding³ are used for an interactive visualization of input waveforms.

To simplify the server-side programs, the file uploading functionality is implemented with File API in HTML5. A text file uploaded by a user is transferred to the servlet as is as an argument of type `String` during a remote procedure call (RPC). Then the string is parsed into arrays of type `double` to hold (t, h_+, h_\times) in each row in the servlet, and “resampled and decimated hierarchically (see Sect. 3).” The servlet invokes a Python script to compute a spectrogram, which is converted to a scalable vector graphics (SVG) file by `gnuplot` and encoded into a Base64 string. A power spectrum is calculated with a Java implementation of fast Fourier transform (FFT)⁴, accompanied by an evaluation of the signal-to-noise ratio (SNR) based on the analytic KAGRA sensitivity curve (Manzotti & Dietz 2012). Note that detector beam-pattern functions are assumed to be unity in the estimation. At the end, the waveform arrays, the Base64 encoded spectrogram, the array for the power spectrum, and the SNR are packed into a single object in JavaScript object notation (JSON), and returned to the web client as the result of the RPC. The waveforms and power spectrum are plotted with Google Charts on the client.

3. Data Reduction Algorithm

By assuming a typical sampling rate of a gravitational wave detector, our programs should be able to handle $N \sim 10^5$ data points on the fly. However, this is a very heavy task for a JavaScript application like SNEGRAF currently. To achieve this goal even on a low-powered CPU, we applied “hierarchical decimation technique” to SNEGRAF. The basic idea of this method is to apply a decimation by a factor of 2 recursively on server side, and to select an adequate result depending on the time duration requested by a user on client side.

1. Find the integer m satisfying $2^m < N \leq 2^{m+1}$ and resample the original waveform evenly into new 2^m points by linear interpolation. This takes $O(m2^m)$ time.
2. Decimate the resampled waveform by 2. This yields new $N_{m-1} = 2^{m-1}$ data points and takes $O(2^{m-1})$ time.
3. Apply the 2nd step recursively until the number of new data points N_i is less than $N_{\text{disp,th}}$ ($= 2048$). At this step, the total amount of data points is exactly less than $N + N/2 + N/4 + \dots = 2N$.
4. On client side, calculate the number of data points $N_{\text{disp},i}$ which fall inside the user specified time range for each decimated data. The total processing time is $O(m^2)$.

¹It was known as Google Web Toolkit previously. <http://www.gwtproject.org/>

²It is also known as Google Visualization API. <https://developers.google.com/chart/>

³GWT Charts. <https://github.com/google/gwt-charts/>

⁴<https://www.nayuki.io/page/free-small-fft-in-multiple-languages>

5. On the client, find the largest i such that $N_{\text{disp},i} \leq N_{\text{disp,th}}$ with a binary search algorithm, and plot the data. This yields new $N_{m-1} = 2^{m-1}$ data points and takes $O(2^{m-1})$ time.
6. Apply the 2nd step recursively until the number of new data points N_i is less than $N_{\text{disp,th}}$ ($= 2048$). At this step, the total amount of data points is exactly less than $N + N/2 + N/4 + \dots = 2N$.
7. On client side, calculate the number of data points $N_{\text{disp},i}$ which fall inside the user specified time range for each decimated data. The total processing time is $O(m^2)$.
8. On the client, find the largest i such that $N_{\text{disp},i} \leq N_{\text{disp,th}}$ with a binary search algorithm, and plot the data.

When N is $\sim 10^5 \approx 2^{17}$, there are just ≈ 150 lookups of the arrays and $\leq N_{\text{disp,th}}$ drawings on the client with just consuming twice as much as the initial memory space. The processing time on client side is reduced by two orders of magnitude thanks to this algorithm, and SNEGRAF quickly responds to the user's operations even on a low-powered computer with an Intel Atom CPU.

4. Future Work

- ☐ A spectrogram and power spectrum displayed on the current version are “static.” We have a plan to make them interactive (e.g., the time range on the input waveform viewer will link to that selected on a spectrogram).
- ☐ To display a sky map, which is a heat map representing the likelihood of the source direction.

Acknowledgments. This work is supported by JSPS KAKENHI Grant Number 17H06364.

References

- Abbott, B. P., Abbott, R., Abbott, T. D., Abernathy, M. R., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., Adhikari, R. X., & et al. 2016, Physical Review Letters, 116, 061102. 1602.03837
- Kotake, K. 2013, Comptes Rendus Physique, 14, 318. 1110.5107
- Manzotti, A., & Dietz, A. 2012, ArXiv e-prints. 1202.4031

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Time in Aladin

Pierre Fernique¹, Daniel Durand², and Ada Nebot¹

¹*Observatoire astronomique de Strasbourg, Université de Strasbourg, CNRS,
UMR 7550, 11 rue de l'Université, F-67000 Strasbourg, France*

²*National Research Council Canada - Herzberg Astronomy and Astrophysics,
Canadian Astronomy Data Centre, Victoria, B.C., Canada*

Abstract. This paper is presenting a few recent Aladin's developments which are designed to handle and display the astronomical time dimension. Aladin (Bonnarel et al. 2000) was originally designed to visualize astronomical data using their spatial coordinates. Using the same basic technology, we have incorporated the time dimension in Aladin.

1. Introduction

Aladin was originally designed to visualize astronomical data based on their spatial coordinates. Using the same technology, we have incorporated a new dimension in Aladin: the time. A new Aladin prototype, based on the core of Version 10, incorporates two new components: a **Time view** window and a **Time coverage** capability. This new Aladin prototype incorporates two new components: a time visualization window and a time coverage capability. The time view window is a light Aladin modification of its regular graphic window originally designed to handle longitude VS latitude plots. This new graphic capability is dedicated to draw scatter plots where the primary axis is time and the secondary axis can be any of the other catalog quantity (i.e. magnitude, flux, radial velocity, etc.). The original spatial view and the new time view are fully interoperable allowing the users to select objects in either views to see them selected in the alternate view. The time coverage capability is based on the technology developed for the Multi-Ordered Coverage (MOCs) (Fernique et al. 2015), replacing the HEALPix space with a time scale instead. Thus the way the user manipulates time coverage is similar to space coverage manipulation, like performing fast coverage intersection or union, generating a time coverage from a list of sources, etc. These new capabilities are already available in the Aladin Beta version available on the Aladin CDS Web site. <http://aladin.u-strasbg.fr>

2. The Time View

The **Time View** window is a simple extension to Aladin's graphic window originally designed to handle longitude VS latitude plots. This new graphic mode is now capable of drawing scatter plots where the primary axis is time and the secondary axis is selected by the user and could use any of the accessible quantities like magnitude, flux, radial

velocity, etc. This new graphic mode is fully interoperable with Aladin’s spatial window so selected objects markers are visible on both windows simultaneously. It is thus possible to explore the time variation of any quantities and identify any interesting measurements and see their localization on the Aladin main image window. Please look at Figure 1

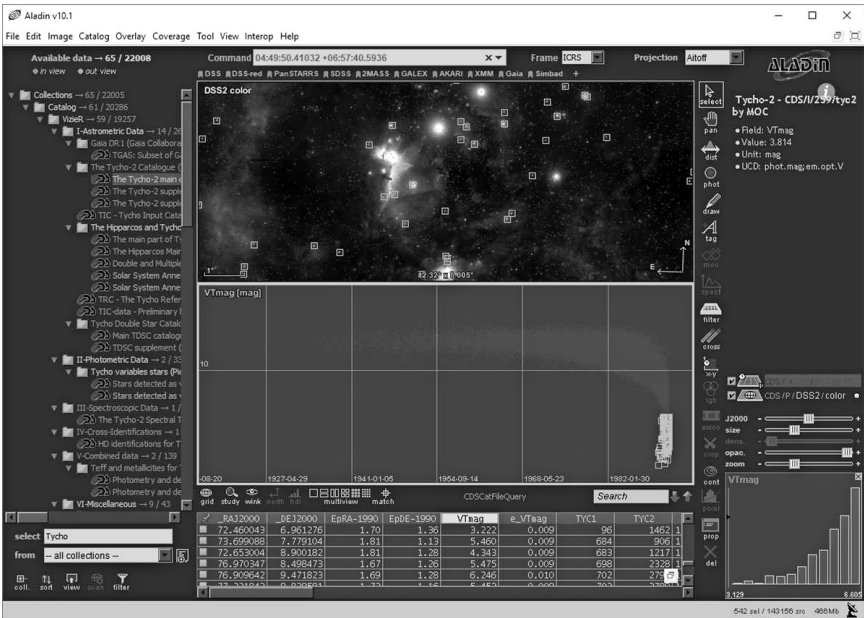


Figure 1. Aladin showing the time view graphic window

3. The Time Coverage

The **Time coverage** capability is based on the technology we used to support the Multi-Ordered Coverage (MOCs). To support the time axis, we simply replaced the HEALPix space discretisation with a time scale using the same properties as the MOC but covering only one axis. (Fernique et al. 2015). With the **Time coverage** support, the user is able to manipulate the time coverage the same way he/she was able to manipulate the space coverage using Aladin. Thus one can perform time coverage manipulation like intersections or unions of different time coverages, generate new time coverage from catalog. For this to be possible, Aladin prototype is introducing a new version of MOC files dedicated for the time axis called T-MOC. Creating T-MOCs was made possible with a very simple modification of the basic MOC java library since it is based on the same algorithm. The T-MOC cell definition are exposed in Table 1.

Visually, Aladin is presenting the T-MOCs like a code bar representing the time coverage of a given collection at a given resolution. It is possible to zoom in and out the T-MOC to explore the time coverage. Please see Figure 2 and Figure 3 which is showing a T-MOC window over Aladin.

Table 1. TMOC cell definition covering 9133y 171d 11h 22m 31.711744s

order	Cell Resolution
0	9133y 171d 11h 22m 31.711744s
1	570y 307d 11h 35m 9.481984s
2	570y 307d 11h 35m 9.481984s
...	...
6	2y 83d 22h 52m 24.177664s
...	...
12	4h 46m 19.869184s
...	...
22	16.384ms
...	...
27	16 μ s
28	4 μ s
29	1 μ s

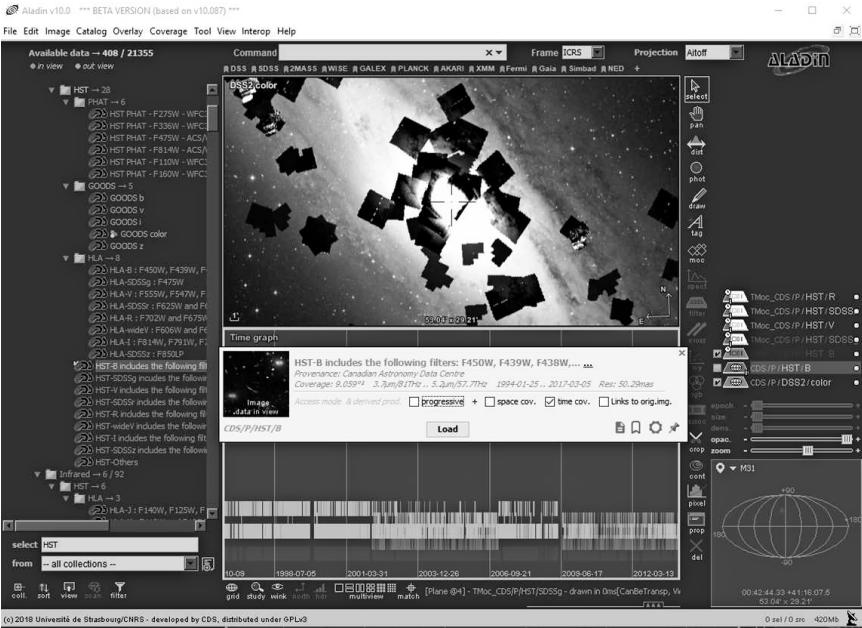


Figure 2. Aladin showing the time view graphic window

4. Requirements

4.1. Requirements for the Time View

In order to be able to display the time values in Aladin, one needs to setup the catalog with its time value or Aladin needs to discover any existing time information. Since at the moment of writing there is no standard way of describing metadata related to time, Aladin is using an heuristic approach to derived the required information for each

catalog. In order to succeed Aladin is using an heuristic approach and try to derive these quantities. As there is multiple parameters like format, scale, offset even the observer location which need to be known, essentially Aladin is executing a best guess approach by scanning all parameters from VOTables (Ochsenbein et al. 2004) or FITS files loaded. When a *discovery* occurs, a little clock symbol is displayed close to the name of the loaded file informing the user of such a discovery. Although this heuristic approach works, it is prompt to errors and it would be much better if a formal description for describing time metadata would exist, e.g. using the *TIMESYS* tag element in VOTable (Please see (Demleitner et al. 2018)).

4.2. Requirements for the T-MOCs

What are the standards we should use to produce T-MOC which are interoperable?

- Using JD(TCB,Barycentric,no offset) requires a Time conversion library.
- Using $1\mu\text{s}$ for order 29 T-MOC resolution covers a period of 9133 years if we use JD=0 (Monday, 4713 B.C. Jan 1, 12:00:00.0) as the zero point.

Please note that for unknown system, the T-MOC will be created at a lower resolution for covering the system imprecision (typically 16minutes).

5. Conclusions

We present two new features of Aladin, both related to Time Domain Astronomy. The first new enhancement of Aladin is the *Time View*, which allows users to visualize light curves, radial velocity plots as well as any other variable of time, while keeping interoperability with the sky position of the objects. The second enhancement of Aladin is the *T-MOC* which allows the users to know the coverage of a catalog in time. These two new features are only possible if the time is well described in terms of its metadata. At the moment Aladin uses a heuristic approach for deriving the needed metadata. But a proper description of metadata in a well defined standard would be very helpful for such a purpose.

References

- Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. 2000, *A&AS*, 143, 33
- Demleitner, A., M.and Nebot, Bonnarel, F., Michel, L., Fernique, P., & Boch, T. 2018, A Proposal for a *TIMESYS* Element in VOTable, IVOA Note 29 October 2018
- Fernique, P., Boch, T., Donaldson, T., Durand, D., O'Mullane, W., Reinecke, M., & Taylor, M. 2015, *ArXiv e-prints*. 1505.02937
- Ochsenbein, F., Williams, R., Davenhall, C., Durand, D., Fernique, P., Giarretta, D., Hanisch, R., McGlynn, T., Szalay, A., Taylor, M. B., & Wicenec, A. 2004, *VOTable Format Definition Version 1.1*, IVOA Recommendation 11 August 2004

Session XI

Multi-Messenger Astronomy

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Coordinating Observations Among Ground and Space-Based Telescopes in the Multi-Messenger Era

Erik Kuulkers,¹ Matthias Ehle,² Carlos Gabriel,² Aitor Ibarra,³ Peter Kretschmar,² Bruno Merín,² Jan-Uwe Ness,² Emilio Salazar,⁴ Jesús Salgado,³ Celia Sánchez-Fernández,⁴ Richard Saxton,⁵ and Emily M. Levesque⁶

¹*European Space Agency/ESTEC, Noordwijk, The Netherlands;*
erik.kuulkers@esa.int

²*European Space Agency/ESAC, Villanueva de la Cañada, Madrid, Spain;*

³*Quasar Science Resources for ESA-ESAC, Villanueva de la Cañada, Madrid, Spain;*

⁴*ATG Europe for ESA-ESAC, Villanueva de la Cañada, Madrid, Spain;*

⁵*Telespazio-Vega UK Ltd. for ESA-ESAC, Villanueva de la Cañada, Madrid, Spain;*

⁶*University of Washington, Seattle, WA USA*

Abstract. The emergence of time-domain multi-messenger (astro)physics requires for new, improved ways of interchanging scheduling information, in order to allow more efficient collaborations between the various teams. Currently space- and ground-based observatories provide target visibilities and schedule information via dedicated web pages in various, (observatory-specific) formats. With this project we aim to: i) standardise the exchange of information about observational schedules and instrument set-ups, and ii) standardise the automation of visibility checking for multiple facilities. To meet these goals, we propose to use VO protocols (ObsTAP-like) to write the services necessary to expose these data to potential client applications and to develop visibility servers across the different facilities.

1. Introduction

Over the last years the scientific demands for simultaneous observations across the electromagnetic spectrum are continuously increasing. This increase has been amplified by the detection of non-electromagnetic events of astrophysical origin, such as high-energy neutrino events, and, in particular, gravitational wave (GW) events. It has culminated with the detection of prompt transient gamma-rays coincident with a GW event caused by the merger of two neutron stars (GRB170817A/GW170817). The latter event involved many facilities on the ground and in space and represented all currently accessible wavelengths (Abbott et al. 2017). Moreover, the transient nature of the event required fast reaction times, in order not to miss any possible ‘afterglow’ emission. With other large-scale facilities coming online soon which will report on transient

events across the EM spectrum (e.g., LSST, SKA), efficient and fast coordination is a must in order to maximize the scientific output.

Although the process to coordinate these observations is currently cumbersome, the demand for coordinated observations is high. For example, of the observations of ESA's space-based facilities INTEGRAL and XMM-Newton, about 10% and 12%, respectively, are coordinated with other observatories (including NuSTAR, HST, Chandra, VLT, Swift). There are nice examples of successful coordinated observations that have produced great and important science results, such as:

- Follow-up observations of the very high-energy neutrino alert on 22 September 2017 by the IceCube Neutrino Observatory, IceCube-170922A. For the first time the source of such an event was found: the event originated from a flaring gamma-ray blazar TXS 0506+056. About 20 ground- and space-based observatories were involved, with about 1010 scientists participating (Aartsen et al. 2018).
- Follow-up observations of the GW event detected by LIGO/Virgo on 17 August 2017 (GW170817). This detection has revolutionized multi-messenger astronomy, as it is the first coincident detection of a gravitational wave in electromagnetic light, i.e., gamma-rays (GRB180817A). Subsequent observations, involving about 70 ground- and space-based observatories, and about 3680 scientists, showed it to be a kilonova, due to the merger of two neutron stars (Abbott et al. 2017).

The coordination we do today, however, is not without difficulties. For example, one of the risks of these mostly ad-hoc collaborations is that the observations are not always strictly simultaneous. If time scales of variability are shorter than the degree of achieved overlap of observations, then the quality of scientific conclusions can be directly impacted. Nowadays there are ongoing efforts (and organizations) aiming to enhance the way we coordinate multi-wavelength and/or multi-messenger astronomy. Some of them are listed here, with no claim of completeness:

- AMON – The Astrophysical Multi-messenger Observatory Network, see <https://www.amon.psu.edu>.
- SCiMMA – Scalable Cyberinfrastructure to support Multi-messenger Astrophysics, see <https://scimma.org>.
- Astronomy ESFRI – Research Infrastructure Cluster (ASTERICS), see <https://www.asterics2020.eu>.
- SmartNet – Simultaneous Multiwavelength Astronomy research in Transients NETwork, see <https://www.isdc.unige.ch/smartnet/> (Middleton et al. 2017).
- DWF – Deeper Wider Faster program, see <http://dwfprogram.altervista.org/> (Andreoni & Cooke 2018).
- TOMs – Target and Observation Managers, see Street et al. (2018).
- Proposal for a multi-messenger institute, see Allen et al. (2018).

There are various steps in the process to improve our coordination process. These are: “getting the alerts”, “automated coordination” and “easier communication”. In the next Sections we go into the subjects in more detail. Note that we mainly concentrate on transient, sudden events, which need quick reaction. Of course, the improved ways of collaboration can be of benefit to other areas in astronomy.

2. Getting the alerts

First, one has to be informed that a transient event is taking place or has just happened. Various ways to communicate all kinds of transients already exist (such as the Astronomers’ Telegrams [ATels], Gamma-ray Coordinates Network [GCN] or Transient Astronomy Network [TAN] Notices, Circulars and Reports, AMON alerts, SuperNova Early Warning System [SNEWS] and Virtual Observatory Events [VOEvents], see below). Receiving notification of an event may be as simple as signing up for these existing feeds/alerts. However, still these various alerts come in various, non-standard, formats.

Moreover, the prioritization of transient follow-up observations will also be a key issue in the next decade. With facilities such as LSST and SKA coming online, as well as the increase in sensitivity of GW and high-energy neutrino facilities, the number of transient detections with an urgent demand for follow-up will skyrocket. Managing the priority and immediacy of these triggers will become a significant challenge. In addition to prioritizing types of observations, a broader prioritization of ‘categories’ of follow-up observations should also be established for each observatory: one has to determine how urgent the follow-up of a given event is and whether other observations - including follow-up observations of another transient - should be interrupted (for example, should a search for the optical counterpart of a GW trigger be interrupted for follow-up imaging of a nearby core-collapse supernova?).

3. Automated coordination & easier communication

When follow-up observations need to be planned, coordination is crucial. Good coordination requires good communication tools, which must be based on standard protocols. It is key to establish good communication channels (i.e., a network) with relevant people between facilities, i.e., Principal Investigators or Project Scientists (those who make decisions about the observations) and observation planners (those that build the observing schedule). E-mail is nowadays the most-used means to communicate, but it is ad-hoc and usually addresses a selected group. One possibility is to use an open, online, messaging and collaboration tool such as Slack (or a dedicated tool like SciApp¹). Users can update the facilities as well as the community, in real-time, or even in advance, of planned and/or executed follow-up observations. The public, in turn, can use the information to better optimize their planned programs.

¹SciApp was designed at ESA/ESAC as a collaborative application, focused to exchange information and knowledge in a specific area using modern, web mobile, technologies. It could make use of protocols (such as ObjVisTAP and ObsLocTAP described in Sec. 4) to gather the information from any astronomical facility and display this information in a user’s friendly way. It maintains all of the discussion and results of a particular observing campaign in one easily accessible area. This application is still in a beta phase and currently on hold. See <https://sciapp.esac.esa.int> for a demonstration.

Another key issue is *rapid* response. Fast transients need fast response times (both in observations and communication). Again, with automation (such as the planning/visibility info; see below) communication becomes more efficient. Based on the available info, a decision tree (e.g., can observations be coordinated with another facility observing at a certain wavelength?) could be used to decide on a go or no-go for follow-up observations. A particular advantage of good communication is in designing ground-based (or space-based) observations that can complement or improve each other's observations. With a fast response time, ground-based (or space-based) observations in the immediate aftermath of a trigger can provide key initial data that can be used in planning additional observations, and ensure rapid acquisition of specific observations where ground-based coverage is equivalent or even superior.

Coordination starts with basic needs. The information that should be made accessible between observatories are of three different types:

- **Observing Schedule:** observations that are already performed with the data already in the relevant data archives are, in most of the cases, accessible to the community. However, the information of the observations that are ongoing or the planned observations are not accessible or only accessible through in-house services (like, e.g., web pages). This information is particularly relevant to allow the follow-up of the observatories operations and plans.
- **Target visibility:** targets are not always visible for a certain observatory. In fact, the information of the periods when a certain target is observable is an output of complex calculations (sometimes geometrical, but, also related with instrument configuration or environment properties). The periods when a particular target could be observable are crucial in order to schedule a coordinated observation. In case of, e.g., variable sources, to ensure that the target is visible in parallel for a certain number of observatories in the expected source high-activity period, allows the planning of relevant science use cases that, in other way, would be impossible.
- **Communication of changes in plans:** changes in the observation planning of facilities are not easy to follow by the community as these are, in many cases, subject to a large number of factors. For example, the decision to change the plans for a particular Target of Opportunity (ToO) could be the outcome of the relevance of the ToO for this observatory, the relevance of the ongoing and near future observations, instrumental aspects, weather, etc.. However, the communication to the community of the decision of the change of the observing plan would be very interesting from the scientific point of view.

There are already some efforts by the astronomical community in place to communicate alerts and ToO follow-ups. In particular, there are initiatives from the International Virtual Observatory Alliance (IVOA) to standardize the reporting of observations from astronomical events. This standard, called VOEvent, uses VOEvent feeds to which scientists and observatories can be subscribed to receive, generate or modify notifications. VOEvent messages are XML documents that include the answers to the observation characterization:

- **<who>** - responsible (author and publisher) of the information contained in the message.

- <how> - instrumental characterization of the observation.
- <what> - links to the data (or measurements) associated with the observation.
- <why> - inferences about the nature of the event.
- <wherewhen> - description of the time and place where the event was recorded.

With this protocol (or a possible extension), the third pillar of observation coordination could be covered.

There are no standards to ensure the information provided are offered to the community in a homogeneous way. An attempt to bring all observation planning information of several observatories together was made in a calendar format: `mySpaceCal.com`. The information displayed in that calendar was obtained from the individual web pages in which observatories publish their planning information. It turned out, however, that the calendar was difficult to maintain. Web page formats change without control, the metadata offered are not homogeneous (no common data model), input parameters are different per observatory and basic information to create a single client is not always present or needs further conversion. It would be better if a standard was agreed with all observatories allowing clients, such as `mySpaceCal.com`, to pull the information from any observatory participating in the consortium.

4. Planning the observations: visibility and planning information

Some degree of coordination for GW follow-up is already de facto in place as a result of joint programs with various observatories (e.g., HST, Chandra, XMM-Newton, Gemini, NRAO). In reality, GW follow-up will also be carried out at a large number of observatories independent of existing programs, and the ability to coordinate effectively between these facilities is necessary in order to maximize the science. The process of long- and short-term planning in general and coordinating observations in particular are becoming more complex in the near future. Automatic elements can make coordination more efficient by cross-correlating visibility and planning information of all involved facilities to generate an optimized observing plan. However, as noted before, currently visibility and planning (past, current, and future) information is not available in a uniform way.

At present there is an effort, led by ESA/ESAC, to define international standards for how observing facilities can make this information available: facilities provide two services in an agreed standard format allowing clients to make queries and receive results in a dedicated format following existing VO (Virtual Observatory) Protocols. This concept has been presented to the IVOA. The implementation of these services could commence rapidly after VO certification, and each facility could build a tool to access the information from the services of all other facilities (simple examples are given in Figs. 1 and 2). Two new protocols are now in the IVOA standardization process: Object Visibility Simple Access Protocol (ObjVisSAP, Sec. 4.1) and Observation Locator Table Access Protocol (ObsLocTAP, Sec. 4.2).

A workshop to discuss this effort was held at ESA/ESAC, Spain on 21 September 2018 in order to discuss the details of the VO protocols, to receive feedback on the proposed standards and to seek collaborators ready to implement prototypes and operational services. There were 29 participants at ESAC and 35 connected by video,

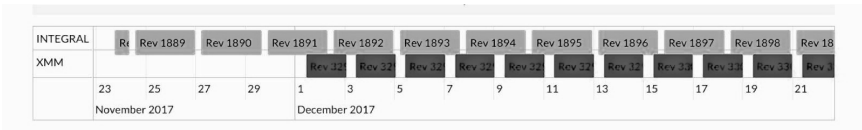


Figure 1. Simple example of visualising visibility constraints from two space-based facilities, i.e., INTEGRAL and XMM-Newton.

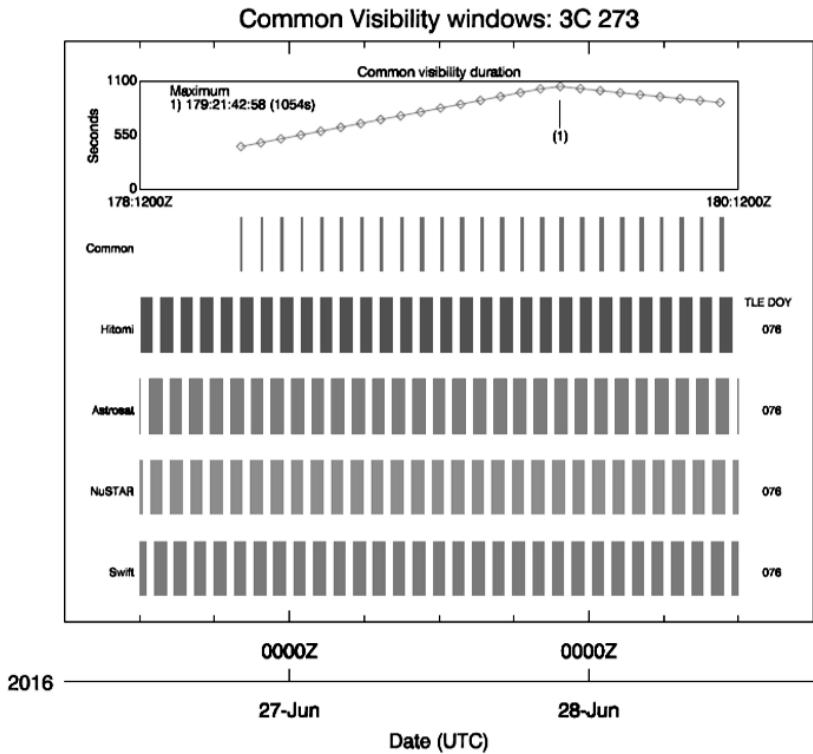


Figure 2. Visualising the optimal coverage (*top*) by using the target visibility information provided by various observatories.

representing more than 40 observatories at all wavelengths, as well as representatives from the GW community, and different multi-messenger/wavelength initiatives (such as SmartNet, ASTERICS). The high turnout demonstrated the broad interest in this initiative and the protocols.

4.1. Object Visibility Simple Access Protocol

The Object Visibility Simple Access Protocol (ObjVisSAP) (Ibarra et al. 2018a) is a simple protocol that allows users to find the periods of time when a particular astronomical target – defined by sky coordinates – is visible. This protocol is an IVOA S*AP protocol, defined in the following way:

- A specific URL for each observatory followed by a simple (standardized) parameter=value interface: the query interface is defined into the protocol allowing clients to ask for a certain target position (Right Ascension [RA] and Declination [Dec] for a certain defined epoch). Also, a qualifier for a certain time range can be provided to constrain the query.
- Table output response: the response of an ObjVisSAP query is an IVOA VOTable document with a set of time periods where the target can be observed by the facility.

ObjVisSAP has been kept very simple in order to facilitate its implementation by as many observatories as possible. However, although the protocol is simple, the implementation of these services by many observatories will allow access to information that is currently cumbersome to extract and allowing the preparation of coordinated proposals between many observatories. Future evolution of this protocol could cover a more complex definition of visibility (e.g., spatial coverage for a certain time), however, always as optional features.

4.2. Observation Locator Table Access Protocol

The Observation Locator Table Access Protocol (ObsLocTAP) (Ibarra et al. 2018b) will allow a common interface for many astronomical facilities to publish the observation planning for current and future observations. It could also cover past observations, but, as noted above, the access to past observations is usually done via the observatories' archives. This protocol is similar to the IVOA ObsTAP protocol (for archived observations) but removing the requirements of links to the data (for obvious reasons) while adding metadata related to the instrumental configuration. Also, the protocol will allow hiding of information of the future observations whenever relevant. For example, in some cases the target to be observed is considered proprietary, so the information provided could be a time slot for a certain observation and a qualifier on the priority to indicate if this observation could be removed in case of an astronomical event. As for ObsTAP, there are some technical details that characterize the protocol:

- Protocol based on TAP: IVOA Tabular Access Protocol (TAP) (Dowler et al. 2010) is an IVOA protocol that publishes the information in a tabular way and provides an access to the metadata in a very close relation to a relational database.
- ADQL language: instead of SQL, the language used to query a TAP service is the IVOA Astronomical Data Query Language (ADQL) (Osuna et al. 2008). ADQL is similar to SQL, but removing data base dependent peculiarities of SQL flavours and adding geometrical functions that simplify queries on the sky. Other languages could be implemented in a TAP service, but ADQL is compulsory to allow the implementation of simple clients.
- Data model: ObsLocTAP services have a common data model, so the same ADQL queries can be sent to the different services by a client and receive compatible/comparable metadata in return. It simplifies the implementation of clients by the requirement of mapping at server side.

5. Conclusions

The era of time-domain and multi-messenger astronomy (when hundreds of astrophysical alerts happen on a daily basis) leads to the need for improved communication and efficient managing of future transient events. New ways of communication need to be set up and (automatic) decision trees need to be put in place, allowing to maximize the scientific output of follow-up observations to transient events. Facilities can enhance their science impact through participation in these processes. We recommend implementing a public and easy-to-use communication system that the community can use to share information about guaranteed, planned, and recently-executed observations. The planning information for these observations should follow existing VO protocols.

Acknowledgments. We thank the participants in the workshop at ESA/ESAC on 21 September 2018 for their engagement, and we hope for a fruitful cooperation and implementation of services using the discussed protocols in the near future.

References

- Aartsen, M. G., Ackermann, M., & Adams, J. *et al.* 2018, *Advances in Space Research*, 62, 2902
- Abbott, B. P., Abbott, R., & Abbott, T. D. *et al.* 2017, *Phys.Rev.D*, 96, 122006. [arXiv:1710.02327](#)
- Allen, G., Anderson, W., & Blaufuss, E. *et al.* 2018, *arXiv e-prints*. [arXiv:1807.04780](#)
- Andreoni, I., & Cooke, J. 2018, *arXiv e-prints*. [arXiv:1802.01100](#)
- Dowler, P., Rixon, G., & Tody, D. 2010, *Table Access Protocol Version 1.0*, IVOA Recommendation 27 March 2010. [arXiv:1110.0497](#)
- Ibarra, A., Salgado, J., Ness, J.-U., Ehle, M., Gabriel, C., Kretschmar, P., Kuulkers, E., Merín, B., Salazar, E., Sánchez, C., & Saxton, R. 2018a, *Object Visibility Simple Access Protocol*, <http://www.ivoa.net/documents/ObjVisSAP/>
- 2018b, *Observation Locator Table Access Protocol*, <http://www.ivoa.net/documents/ObsLocTAP/>
- Middleton, M. J., Casella, P., & Gandhi, P. *et al.* 2017, *New Astronomy Reviews*, 79, 26. [arXiv:1709.03520](#)
- Osuna, P., Ortiz, I., Lusted, J., Dowler, P., Szalay, A., Shirasaki, Y., Nieto-Santisteban, M. A., Ohishi, M., O’Mullane, W., VOQL-TEG Group, & VOQL Working Group. 2008, *IVOA Astronomical Data Query Language Version 2.00*, IVOA Recommendation 30 October 2008. [arXiv:1110.0503](#)
- Street, R. A., Bowman, M., Saunders, E. S., & Boroson, T. 2018, in *Software and Cyberinfrastructure for Astronomy V*, vol. 10707 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 1070711. [arXiv:1806.09557](#)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Searching for Optical Counterparts to Gravitational Wave Events with the Catalina Sky Survey

Michael J. Lundquist¹, David Sand¹, Eric Christensen², Wen-fai Fong³, and Kerry Paterson³

¹*University of Arizona/Steward Observatory, Tucson, AZ, USA;*
mlundquist@email.arizona.edu

²*University of Arizona/Lunar Planetary Laboratory, Tucson, AZ, USA*

³*Northwestern University, Evanston, IL, USA*

Abstract. The recent detection of the optical counterpart to the gravitational wave event GW170817 has ushered in the new era of multi-messenger, gravitational wave astronomy. On 17 August 2017, the gravitational wave resulting from a binary neutron star merger was detected by the Advanced Laser Interferometer Gravitational-wave Observatory (Abbott et al. 2009, LIGO) and Advanced Virgo Observatory (Acernese et al. 2015, Virgo). Shortly after the gravitational wave was detected, telescopes from around the world searched for and then studied the kilonova associated with the merger. Here, we outline our software and strategy for discovering optical counterparts to future gravitational wave events using data from the Catalina Sky Survey.

1. Catalina Sky Survey

The Catalina Sky Survey (CSS) is a NASA funded project dedicated to the discovery and tracking of near Earth objects. CSS uses three telescopes in the Santa Catalina Mountains near Tucson, Arizona. These telescopes include 1.5 m and 1.0 m telescopes on the summit of Mt. Lemmon as well as a 0.7 m telescope on Mt. Bigelow. CSS telescopes operate 24 nights per lunar cycle with a break near the full moon Christensen et al. (2016).

Currently, our search for optical counterparts uses the 1.5 m telescope, but future plans including using the 0.7 m telescope as well. Survey operations at the 1.5 m telescope deliver a limiting magnitude of $V \sim 21.5$ mag in 30s exposures for a 5.0 deg^2 field of view. This allows the survey to cover 1000 deg^2 each night. The 0.7 m telescope delivers a limiting magnitude of $V \sim 19.5$ mag in 30s exposures for a 19.4 deg^2 field of view covering 4000 deg^2 per night. This combination of depth and breadth will be an asset to our search for optical counterparts.

2. Observational Strategy

We have developed software to trigger CSS observations automatically. The LIGO/Virgo alerts for gravitational wave events are distributed using the Gamma-ray Coordinates Network/Transient Astronomy Network (GCN/TAN) and include information

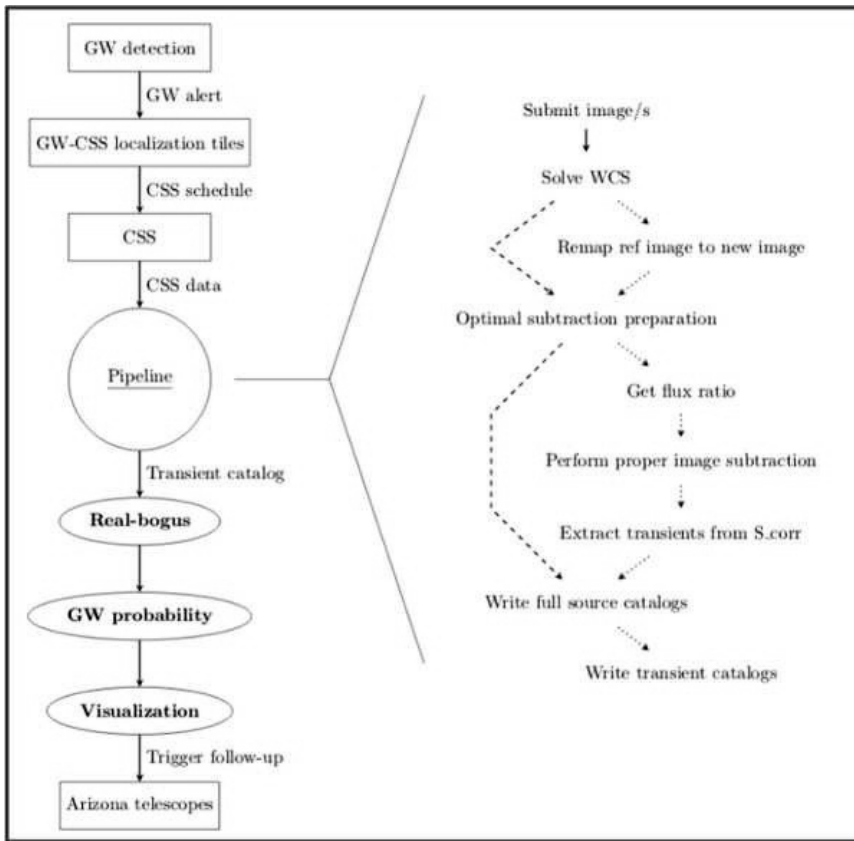


Figure 1. Our software workflow. The pipeline receives the CSS data in real time and uses previously observed reference images to perform an optimal subtraction of the images using the ZOGY algorithm and provide catalogs of transient candidates.

about the localization, false alarm rate, and binary neutron star probability among other parameters. Once we have received the alert, we filter for a low false alarm rate and high binary neutron star probability and store the alert in a PostgreSQL database. We then use the localization to determine which standard CSS fields have the highest probability of containing the gravitational wave optical counterpart and are observable. These fields are saved to a file that can be automatically loaded into the CSS queue scheduler and an email is sent to notify the observers of the event. These fields get incorporated into the regular CSS queue at a higher priority level so that they will be observed first, but the triggering of observations is otherwise transparent to the observer. Our observational strategy piggybacks on the Catalina Sky Survey search for near earth asteroids with minimal impact to their ongoing survey operations.

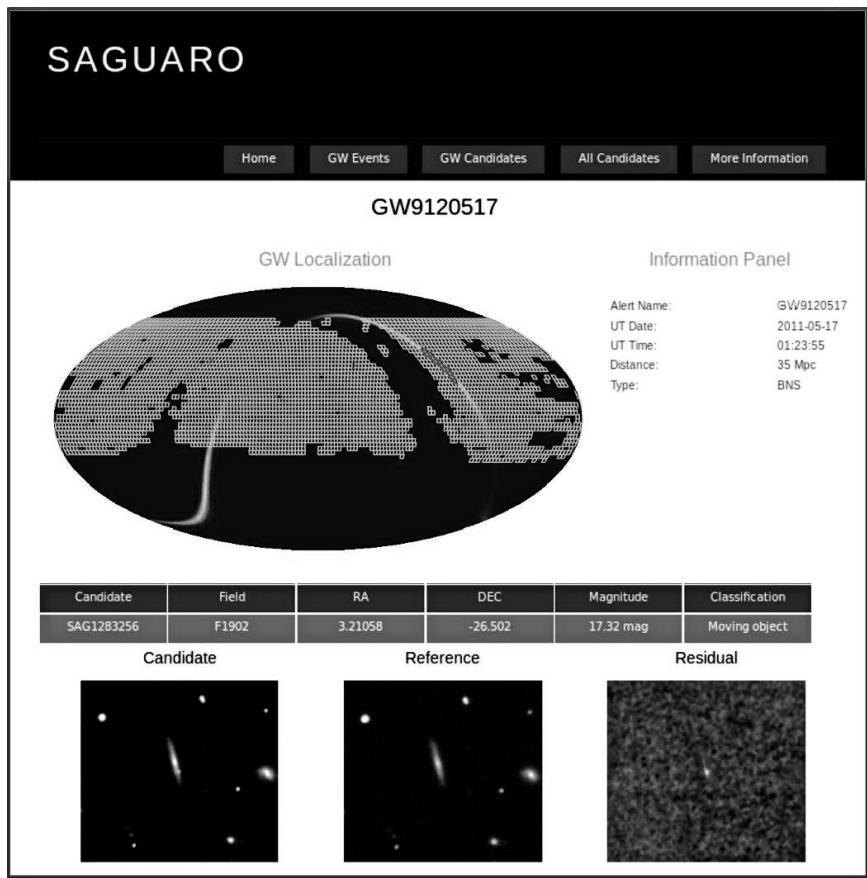


Figure 2. For each alert, information from the alert, the CSS fields that have been triggered overlaid on the localization probability map, and image cutouts for each candidate from the pipeline are shown for human vetting.

3. Transient Detection Pipeline

We use the reduction pipeline developed for MeerLICHT and BlackGEM Bloemen et al. (2016) to analyze the CSS data in real time to detect transients. We use a PostgreSQL database to keep track of which fields are triggered for gravitational wave events and prioritize them in the transient detection pipeline. A workflow of our software is shown in Figure 1. The pipeline uses previously observed reference images to perform an optimal subtraction of the image and provide catalogs of transient candidates. The pipeline also performs basic reduction steps, such as de-biasing, flat fielding, gain correction, cosmic ray cleaning, and bad pixel masking for each image.

The image subtraction part of the MeerLICHT pipeline makes use of the image subtraction method by (Zackay et al. 2016, ZOGY). ZOGY uses statistical principles to derive the optimal statistic for transient detection. Instead of a convolution kernel, ZOGY uses the PSF across the reference and new images during the photometric

alignment, typically providing an improved representation of the sources in the image compared to the convolution kernel fit using HOTPANTS (Becker 2015). ZOGY uses the derived PSFs, along with an estimate of the background standard deviation and flux ratios, to perform the image subtraction by using Fast Fourier Transforms. The difference image is then used to create a significance and a corrected significance image representing the significance of all pixels in the difference image, with the corrected significance image including corrections for source noise and astrometric error. These candidates are stored in the PostgreSQL database.

4. Candidate Vetting

The transient detection pipeline provides image cutouts for each candidate transient. We have developed an internal website for candidate vetting and early analysis. This website runs on a Flask web application and uses Python to query the PostgreSQL database containing information about the LIGO/Virgo Alert, the CSS fields that were triggered, and the pipeline data products. Figure 2 shows an example gravitational wave candidate page from an early version of the website. For each alert, this internal webpage shows information from the alert, a plot of the localization probability map with the CSS fields overlaid, and lists of candidates from the pipeline. Once the candidates have been vetted to remove false positives due to cosmic rays, moving objects, or variable stars, a list of qualified candidates will be made available to the public.

References

- Abbott, B. P., Abbott, R., Adhikari, R., Ajith, P., Allen, B., Allen, G., Amin, R. S., Anderson, S. B., Anderson, W. G., Arain, M. A., & et al. 2009, Reports on Progress in Physics, 72, 076901. 0711.3041
- Acernese, F., Agathos, M., Agatsuma, K., Aisa, D., Allemandou, N., Allocca, A., Amarni, J., Astone, P., Balestri, G., Ballardin, G., & et al. 2015, Classical and Quantum Gravity, 32, 024001. 1408.3978
- Becker, A. 2015, HOTPANTS: High Order Transform of PSF AND Template Subtraction, Astrophysics Source Code Library. 1504.004
- Bloemen, S., Groot, P., Woudt, P., Klein Wolt, M., McBride, V., Nelemans, G., K rding, E., Pretorius, M. L., Roelfsema, R., Bettonvil, F., Balster, H., Bakker, R., Dolron, P., van Elteren, A., Elswijk, E., Engels, A., Fender, R., Fokker, M., de Haan, M., Hagoort, K., de Hoog, J., ter Horst, R., van der Kevie, G., Kozłowski, S., Kragt, J., Lech, G., Le Poole, R., Lesman, D., Morren, J., Navarro, R., Paalberends, W.-J., Paterson, K., Pawłaszczek, R., Pessemier, W., Raskin, G., Rutten, H., Scheers, B., Schuil, M., & Sybilski, P. W. 2016, in Ground-based and Airborne Telescopes VI, vol. 9906, 990664
- Christensen, E. J., Carson Fuls, D., Gibbs, A., Grauer, A., Johnson, J. A., Kowalski, R., Larson, S. M., Leonard, G., Matheny, R., Seaman, R. L., & Shelly, F. 2016, in AAS/Division for Planetary Sciences Meeting Abstracts #48, vol. 48 of AAS/Division for Planetary Sciences Meeting Abstracts, 405.01
- Zackay, B., Ofek, E. O., & Gal-Yam, A. 2016, ApJ, 830, 27. 1601.02655

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

CALET Gamma-ray Burst Monitor Web-analysis System

Ken Ebisawa,¹ Satoshi Nakahira,² Takanori Sakamoto,³ and
Atsumasa Yoshida³

¹*ISAS/JAXA, Sagamihara, Kanagawa, 25-25210, Japan; ebisawa@isas.jaxa.jp*

²*RIKEN, Wako, Saitama, 351-0198, Japan*

³*Aoyama Gakuin University, Sagamihara, Kanagawa, 252-5258, Japan*

Abstract. CALET (CALorimetric Electron Telescope) has been installed and operational on the Japanese Experiment Module Exposed Facility of the International Space Station (ISS) since August 2015. We describe the Web analysis system for the CALET Gamma-ray Burst Monitor (CGBM), which is publicly available from DARTS.

1. Introduction

CALET, CALorimetric Electron Telescope (Torii & Calet Collaboration 2015), was launched in August 2015 and installed on the Japanese Experiment Module-Exposed Facility (JEM-EF) of the International Space Station. CALET carries two observational instruments, the CALorimeter (CAL), whose main target is cosmic-rays, and the CALET Gamma-ray Burst Monitor (CGBM; Yamaoka et al. (2013)).

CGBM is composed of a single Soft Gamma-ray Monitor (SGM) and two Hard X-ray Monitors (HXM1 and HXM2). Their energy bands are ~ 100 keV to ~ 20 MeV (SGM) or ~ 7 keV to ~ 1 MeV (HXM), and the fields of view are the \sim whole sky excluding earth (SGM) or \sim one steradian (HXM). The CGBM public data are released from DARTS (<http://darts.isas.jaxa/astro/calet>) in the standard FITS format, immediately after the pipeline processing.

2. CGBM Archive Data

CGBM produces two types of scientific data; one is the monitoring histogram data which is always output, and the other is the event-by-event data output only when triggered. The monitor data have two types of the histograms, Time History (TH) data with a 1/8 sec time-resolution and 8 energy channels, and Pulse Height (PH) data with a 4 sec time-resolution and 512 energy channels. The event data have a time-resolution of $62.5 \mu\text{s}$ and digitized into 4096 channels. As of December 2018, only the monitor data are archived and publicly available from DARTS.

All the monitor data are simultaneously taken with low gain (LG) and high-gain (HG) to cover different energy ranges with different resolutions. As a consequence, 12 histogram files are produced for a single observation (combination of the three sensors, TH and PH, LG and HG). In the CGBM data archive, data are split for each obser-

vation date. For instance, for the observation on December 16, 2018, the following 12 files are available under <http://darts.isas.jaxa.jp/pub/calet/cgbm-v1.0/obs/2018/20181216/monitor/>:

cgbm_20181216_hxm1_hg.ph	cgbm_20181216_hxm1_hg.th
cgbm_20181216_hxm1_lg.ph	cgbm_20181216_hxm1_lg.th
cgbm_20181216_hxm2_hg.ph	cgbm_20181216_hxm2_hg.th
cgbm_20181216_hxm2_lg.ph	cgbm_20181216_hxm2_lg.th
cgbm_20181216_sgm_hg.ph	cgbm_20181216_sgm_hg.th
cgbm_20181216_sgm_lg.ph	cgbm_20181216_sgm_lg.th

3. CGBM web-analysis tool

Users may download these FITS archive data from DARTS and can analyze using their own tools. Meanwhile, we are developing a user-friendly CGBM web-analysis tool (<http://darts.isas.jaxa.jp/astro/calet/cgbmweb/LCViewer.html>), such that users can select, browse, and quickly analyze the CGBM data only using a web-browser without downloading the data. The tool is mostly written in Python and C++, and makes use of public libraries such as Armadillo, HEALPix, Boost, PROJ, GEOS, basemap, and in-house tools developed for MAXI (Matsuoka et al. 2009).

In the initial screen, users specify the time-range and bin-size, instruments and energy bands to display the burst light-curve (Figure 1). When the light-curve is displayed, users select the time-ranges for the burst (green) and the background (yellow), and click the “build product” button. Then, the system calculates the background subtracted spectrum (Figure 2). The source and background spectral files are made in the xspec format, which users can download together with the instrumental response for spectral analysis. In addition, observing field-of-view (Figure 3) and the projected ISS orbit at the time selection (Figure 4) are calculated.

4. Future plans

The current version 1 archive is only available from DARTS. In the future, we plan to release the event data, and improve the data format and the directory structure to enhance interoperability. We are currently working with HEASARC at NASA/GSFC, so that future CGBM data are also released from HEASARC.

Acknowledgments. We thank all the CALET team members and the DARTS members for their supports in the CGBM data archive and the web-analysis tool developments.

References

- Matsuoka, M., Kawasaki, K., Ueno, S., Tomida, H., Kohama, M., Suzuki, M., Adachi, Y., Ishikawa, M., Mihara, T., Sugizaki, M., Isobe, N., Nakagawa, Y., Tsunemi, H., Miyata, E., Kawai, N., Kataoka, J., Morii, M., Yoshida, A., Negoro, H., Nakajima, M., Ueda, Y., Chujo, H., Yamaoka, K., Yamazaki, O., Nakahira, S., You, T., Ishiwata, R., Miyoshi, S., Eguchi, S., Hiroi, K., Katayama, H., & Ebisawa, K. 2009, PASJ, 61, 999. 0906.0631
- Torii, S., & Calet Collaboration 2015, in 34th International Cosmic Ray Conference (ICRC2015), edited by A. S. Borisov, V. G. Denisova, Z. M. Guseva, E. A. Kanevskaya,

CALET CGBM on-demand

Light curve viewer on demand

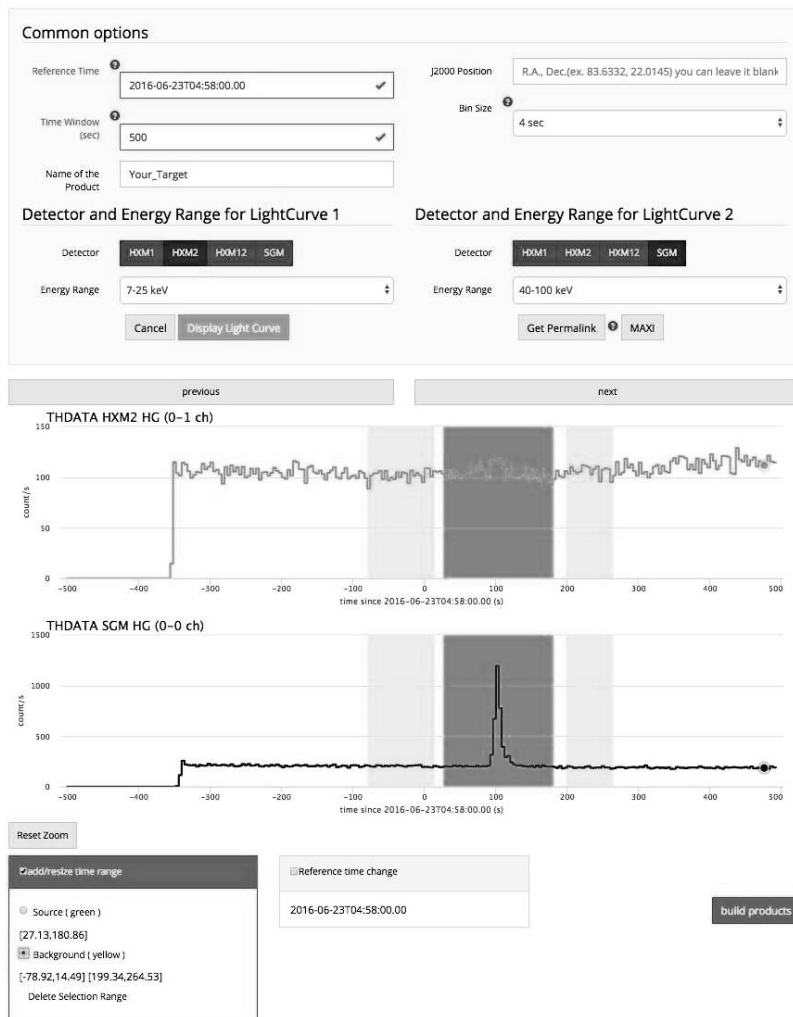


Figure 1. Display of the burst light curve in the two energy bands, and user-selection of the burst-period (green) and the background-period (yellow).

M. G. Kogan, A. E. Morozov, V. S. Puchkov, S. E. Pyatovsky, G. P. Shoziyoev, M. D. Smirnova, A. V. Vargasov, V. I. Galkin, S. I. Nazarov, & R. A. Mukhamedshin, vol. 34 of International Cosmic Ray Conference, 581

Yamaoka, K., Yoshida, A., Sakamoto, T., Takahashi, I., Hara, T., Yamamoto, T., Kawakubo, Y., ota Inoue, R., Terazawa, S., Fujioka, R., Senuma, K., Nakahira, S., Tomida, H., Ueno, S., Torii, S., Cherry, M. L., Ricciarini, S., & the CALET collaboration 2013, arXiv

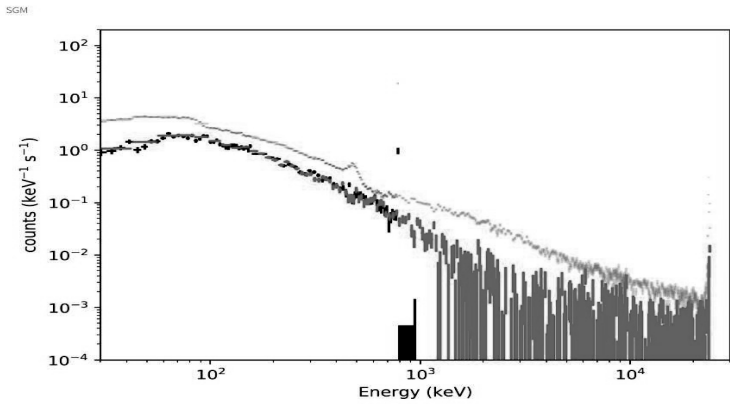


Figure 2. The background subtracted burst energy spectra.

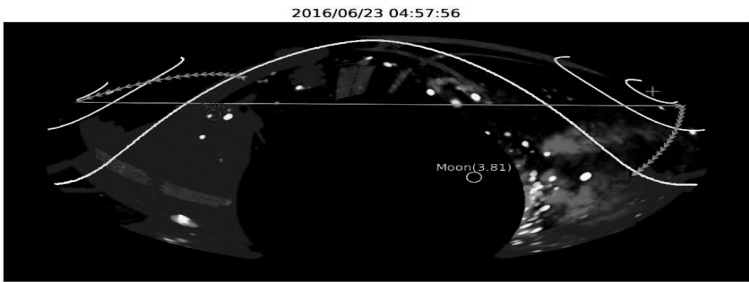


Figure 3. The CGBM field of view, shown on the MAXI image.

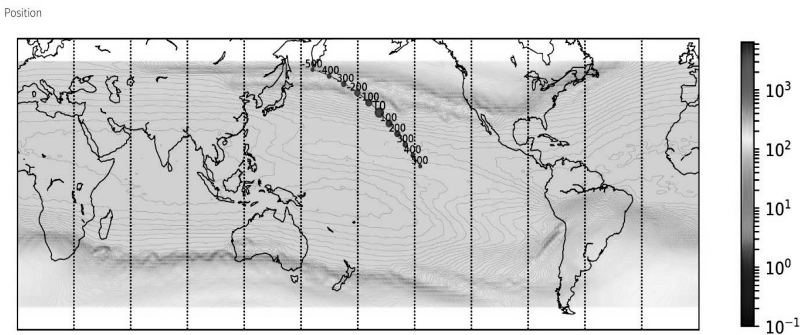


Figure 4. Projected ISS orbit at the time selection.

e-prints. 1311.4084

Session XII

Algorithms

An Overview of the LSST Image Processing Pipelines

James Bosch^{1,2}, Yusra AlSayyad², Robert Armstrong³, Eric Bellm⁴,
Hsin-Fang Chiang⁵, Siegfried Egg1⁴, Krzysztof Findeisen⁴,
Merlin Fisher-Levine⁶, Leanne P. Guy⁶, Augustin Guyonnet⁷, Željko Ivezić⁴,
Tim Jenness⁶, Gábor Kovács⁴, K. Simon Krughoff⁶, Robert H. Lupton²,
Nate B. Lust², Lauren A. MacArthur², Joshua Meyers³, Fred Moolekamp^{2,8},
Christopher B. Morrison⁴, Timothy D. Morton^{2,9}, William O'Mullane⁶,
John K. Parejko⁴, Andrés A. Plazas², Paul A. Price², Meredith L. Rawls⁴,
Sophie L. Reed², Pim Schellart², Colin T. Slater⁴, Ian Sullivan⁴,
John. D. Swinbank⁴, Dan Taranu², Christopher Z. Waters², and
W. M. Wood-Vasey¹⁰

¹*jbosch@astro.princeton.edu*

²*Princeton University, Princeton, NJ, U.S.A.*

³*Lawrence Livermore National Laboratory, Livermore, CA, U.S.A.*

⁴*University of Washington, Seattle, WA, U.S.A.*

⁵*National Center for Supercomputing Applications, Urbana, IL, U.S.A.*

⁶*LSST Project Management Office, Tucson, AZ, U.S.A.*

⁷*Harvard University, Cambridge, MA, U.S.A*

⁸*Rider University, Lawrenceville, NJ, U.S.A.*

⁹*University of Florida, Gainesville, FL, U.S.A.*

¹⁰*University of Pittsburgh, Pittsburgh, PA, U.S.A.*

Abstract. The Large Synoptic Survey Telescope (LSST) is an ambitious astronomical survey with a similarly ambitious Data Management component. Data Management for LSST includes processing on both nightly and yearly cadences to generate transient alerts, deep catalogs of the static sky, and forced photometry light-curves for billions of objects at hundreds of epochs, spanning at least a decade. The algorithms running in these pipelines are individually sophisticated and interact in subtle ways. This paper provides an overview of those pipelines, focusing more on those interactions than the details of any individual algorithm.

1. Introduction

Over the course of the 2020s, the Large Synoptic Survey Telescope (Ivezić & LSST Collaboration 2008) will produce a petabyte-scale astronomical dataset with an unprecedented combination of depth, area, and time-domain sensitivity across six opti-

cal/near-infrared bands. The LSST project is much more than a telescope; it is first and foremost a survey, and one accompanied by an extensive data management effort. LSST Data Management (Jurić et al. 2017; O’Mullane & LSST Data Management Team 2018) is responsible for producing many different data products as well as providing services to archive and serve the data to the community. This paper provides an overview of the pipelines and algorithms responsible for generating those data products, including high-level, “science-ready” catalogs.

The LSST processing pipelines encompass two major components. The Prompt Processing¹ pipelines will run in near-real-time, generating alerts for transient detections from image differencing within 60 seconds of their observation, as well as subsequent forced photometry and orbit updates for solar system objects over the course of the next 24 hours. The Data Release pipelines will run on a yearly cadence², producing a complete reprocessing of the full survey dataset each time. The Data Release pipelines generate calibrated images, coadds, image differences, and catalogs of detections and measurements derived from all of these.

The catalogs also split into two general categories. In the LSST nomenclature, *object* catalogs have a single entry for each astrophysical object, which in general aggregates measurements from multiple observations. *Source* catalogs have different entries for each observation of an object. Multiple catalogs in both categories are produced in both Prompt Processing and Data Release production.

The inclusion of image differencing in both the Data Release pipelines and the Prompt Processing pipelines is worth drawing attention to, as it highlights the fact that the algorithms developed by LSST cannot be cleanly split into Prompt and Data Release components. Both groups of pipelines are built on top of a common algorithmic codebase and middleware system (e.g. Jenness et al. 2019), and each Data Release processing campaign will re-do almost everything done by the previous year’s³ Prompt processing.

The LSST pipelines and algorithms have been in development for more than a decade; they are very much functional, but they are by no means complete. This paper is intended to summarize their expected state in early operations. For a much more thorough description of many of these algorithms in their current state, we refer the reader to Bosch et al. (2018), which describes the processing of the Hyper Suprime-Cam Strategic Survey Program (Aihara et al. 2018) using software derived from a recent version of the LSST codebase. Project documents also provide additional information on planned LSST data products (LSE-163; Jurić et al. 2018) and algorithms (LDM-151; Swinbank et al. 2017).

¹ “Prompt Processing” is now preferred over the “Alert Production” terminology LSST has used in the past, reflecting the fact that these pipelines generate more than just alerts.

²The first two data releases are currently planned to be only six months apart.

³ This is a slight simplification of the timing; the set of raw observations included in an LSST Data Release is frozen before processing starts. That processing will take the better part of a year, so it may be more than a year before a particular observation first appears in a data release.

2. Goals and Philosophy

Official survey pipelines like those being developed for LSST must be designed with an extremely broad range of scientific use cases in mind. This can actually make the goals and priorities of survey pipeline development quite different those of algorithm development in the pursuit of more specific science goals, despite similarities in methodology.

The outputs of an official survey pipeline are expected to act as proxies for the raw data; they should be as free of assumptions, filtering, and biases as possible. Avoiding any kind of modeling is of course impossible: catalogs are themselves models of the sky as a set of discrete objects, built on top of models of the observatory and atmosphere. Ideally survey pipelines employ multiple models, deferring the selection of the most appropriate model to downstream analyses. For example, we measure the fluxes of each object under both the assumption that it is a point source and the assumption that it is a galaxy, rather than classifying it first and using that classification to determine how to photometer it. Given the diversity of astrophysical objects and science goals, this naturally leads to a proliferation of measurements, which may seem at first to be a waste of processing time and storage, given that many astronomical objects can be securely classified to a degree that makes some measurements obviously inappropriate. In practice, however, the computational and storage demands of a survey are driven by its faintest and most poorly-resolved populations, where classifications are rarely secure.

Multiple models are unfortunately not an option in the early stages of pipeline processing; we simply cannot afford to fit multiple *image characterization* models, such as the point-spread function (PSF) or the sky background, given that utilizing different versions of those models in downstream processing would lead to a combinatorial explosion in the catalog size. Instead, the quality of these models must meet the requirements of the most demanding downstream science, and in some cases, those requirements are in tension. Astronomers interested in faint point sources, for example, prefer local background estimates even when this conflates sky backgrounds with smooth, low-surface-brightness features from nearby extended objects, while those interested in those low-surface-brightness objects require backgrounds to be measured only on very large spatial scales. A survey pipeline that meets the requirements of both science cases must thus use a better overall background model than either science case would require independently.

Survey pipelines also typically operate under stringent computational and storage constraints, however. To make the above “deferred-model-selection” and “best-possible-image-characterization-models” philosophies computationally tractable, survey pipelines such as those used in the SDSS (*Photo*; Lupton et al. 2001) and Pan-STARRS (*IPP*; Waters et al. 2016) traditionally have made two major simplifying assumptions:

- Astronomical objects are sufficiently separated on the sky to be detected and well-measured independently (even if this involves an explicit *deblending* step to further separate them).
- Calibrations such as the PSF and background can be characterized well enough that their uncertainties can be neglected when computing the overall errors on per-object measurements (even if their biases cannot be).

The applicability of these assumptions and trade-offs has largely been borne out in the science results. Bayesian methods that relax these assumptions by jointly sampling or optimizing multi-object likelihoods have yet to produce results that improve on traditional pipelines in more than very limited respects, despite being orders of magnitude more computationally expensive (Regier et al. 2016; Brewer et al. 2013; Schneider et al. 2015); in many cases, they are still too expensive for fair comparisons to even be made.

This may change in the era of LSST, however, because the survey's depth comes with a dramatic increase in object density, and its size (and accompanying reduction in statistical errors) tightens requirements on systematic errors. The SDSS deblending algorithm in particular has already been demonstrated to be inadequate at LSST depths (Bosch et al. 2018), and it remains unclear whether an improved algorithm with essentially the same philosophical approach can solve this problem. Methods based more directly on joint or iterative fitting of multi-object likelihoods (e.g. Melchior et al. 2018; Barden et al. 2012; Drlica-Wagner et al. 2018) probably have a role to play as well, but defining constraints or priors that are informative enough to ensure efficient convergence without biasing derived measurements is a significant challenge.

3. Prompt Pipelines and Data Products

3.1. Single-Epoch Processing

Processing of LSST science observations begins with Instrument Signature Removal (ISR), which includes basic detrending (flat-fielding, bias subtraction, fringe correction, etc), nonlinearity and crosstalk correction, and masking of bad and saturated pixels. A full description of LSST's photometric and astrometric calibration plans is well beyond the scope of this paper, but it is worth noting that the flat applied here is a fairly sophisticated construct, derived from data from many different sources, including:

- traditional dome flats, to constrain small-scale quantum efficiency (QE) and pixel-size variations;
- a collimated beam projector with a tunable laser (Coughlin et al. 2016), to constrain wavelength dependence, nonuniform illumination, and scattered light in the dome flats;
- an auxiliary telescope equipped with a low-resolution spectrograph, to constrain atmospheric transmission;
- models fit to dithered observations of stars ("star flats"), to constrain degeneracy's between QE and pixel size variation and provide an independent constraint on the illumination correction (via methodology similar to that of Bernstein et al. 2017b).

In ISR, our goal is to apply a flat that transforms the raw science image into one with *surface brightness* pixels that is photometrically flat for objects with the spectral energy distribution (SED) of the sky. Such an image is inappropriate for precision photometry, but ideal for background subtraction.

The next few steps all fall into the category of single-epoch direct image characterization, in which we build models that describe the state of the observational system

and how it transforms the true sky into the image that we see. This includes background subtraction, PSF modeling, finding and interpolating over cosmic rays, measuring and applying aperture corrections, detecting, deblending, and measuring sources. Each of these steps can only be done after at least one of the others, so in practice we'll have to repeat some of them as we iteratively improve the models they produce.

The detection, deblending, and measurement steps are essentially the same as those run in the SDSS *Photo* pipeline (Lupton et al. 2001):

- Detection is responsible for identifying above-threshold regions and peaks within them; when multiple peaks appear in the same region, we call this a blend.
- Deblending creates a *child* record for each peak in a blend, along with an image that contains the best estimate of the flux from just that child.
- Measurement applies a sequence of plug-in algorithms (e.g. centroiding, aperture photometry, PSF photometry, shapes) to both the *parent* (the original image of each blend, hence interpreting it as a single object) and the child images produced by the deblender.

These are described much more fully in Bosch et al. (2018). As noted previously, we will probably need a new approach to deblending when processing deep (*i.e.*, coadded) LSST images, but at single-epoch depths the SDSS algorithm still performs adequately.

Once we have produced a reliable source catalog, we match to a reference catalog (produced in the most recent data release⁴) to photometrically and astrometrically calibrate the image. This first requires correcting the photometry for our background-optimized flat-fielding. This could be done by dividing by our original flat-field image and multiplying by a new one that transforms to flux-valued pixels and flattens an SED more typical of astrophysical sources than the sky, prior to our final source measurement iteration. In practice, we expect to be able to apply the same correction to sufficient accuracy in catalog-space after measurement.

3.2. Image Differencing

The heart of the Prompt pipelines is image subtraction and transient detection on the differences. Each science observation is subtracted from a template coadd (also produced in the most recent data release) covering the same area of sky. We first resample the template to the pixel coordinate system of the new science image, and then convolve it with a kernel to match its PSF to the PSF of the new observation. Our baseline algorithm (Reiss & Lupton 2016) for computing this kernel is primarily based on that of Alard & Lupton (1998), but incorporates some ideas from Zackay et al. (2016) as well. The former maximizes the signal-to-noise of point-source detections in the limit that the template is noiseless, but does not rely on having accurate PSF models (which cannot be obtained in general in crowded fields). The latter maximizes the signal-to-noise even when both images have noise but requires a high-quality Fourier-domain ratio of the template and science image PSFs as an input. Our hybrid method should have the advantages of both approaches.

LSST lacks an atmospheric dispersion corrector, so differential chromatic refraction (DCR) makes the effective PSF (and hence the difference kernel) for each source a

⁴Plans for prompt processing prior to DR1 are still to be determined.

strong function of its SED. This presents a significant challenge to all existing methods for image subtraction, which assume the difference kernel is a function only of position on the image. Our baseline algorithm for dealing with DCR (Sullivan 2018) involves constructing “sub-band” templates that can be combined with different weights for each source during image differencing. The algorithm is still at a prototype stage, but early results are promising.

Once image subtraction is complete, we expect to run essentially the same detection algorithm we apply to single-epoch direct images to obtain *DIASources* (Difference Imaging Analysis Sources). Deblending on difference images is an easier problem than even deblending on shallow single-epoch direct images, because galaxies are static and should subtract cleanly, leaving only point sources, dipoles, and trailed sources that are significantly easier to model. Many of our direct-image measurement algorithms can also be applied directly to difference images, though we will run additional measurement algorithms relevant only for moving objects.

Details of the measurements included in the *DIASource* table (as well as all other LSST data products) can be found in the project’s Data Product Definition Document (LSE-163; Jurić et al. 2018).

3.3. Association and Alerts

Before alerts are issued, new *DIASources* are spatially matched with the *DIAObject* and *SSObject* (Solar System Object) tables, providing extra context to include in the alert. Entries in these tables are themselves constructed from unassociated *DIASources* (in the case of *DIAObject*) and sets of linked *DIASources* that are consistent with solar system orbits (*SSObjects*). New *DIAObjects* are created and existing *DIAObjects* are updated immediately after *DIASource* measurement, while *SSObject* linkage and orbits are re-analyzed during the 24-hour period following observation.

We also perform *precovery* on all new *DIAObjects* by performing forced photometry at their positions in all difference images observed in the previous 30 days, in order to provide light-curves for new transients before they rise above our detection threshold. Precovery is also run during the 24 hours following the observation of the object.

Alert packets will be sent out 60 seconds after each observation for all *DIASource* measurements obtained in that observation. The packets will include the *DIASources*, their associated *DIAObjects* or *SSObjects*, any associated *DIASources* from the last 12 months, and identifiers for several nearby Objects from the most recent Data Release Processing. The Prompt Products Database (PPDB) holding the *DIASource*, *DIAObject*, and *SSObject* tables will also provide an interface to this information; it will be queryable about 24 hours after the observation.

4. Data Release Pipelines and Data Products

4.1. Image Characterization and Calibration

The Data Release pipelines begin with essentially the same single-epoch processing that is run in the Prompt pipelines. Each LSST Data Release is intended to be entirely independent, however, so the LSST-produced reference catalogs and templates used in Prompt processing are not an option. Instead, we plan to match and calibrate to the Gaia catalog (Gaia Collaboration et al. 2016), but in the Data Release pipelines this just provides a preliminary calibration. For the final calibration, we will move away

from processing each observation independently and instead fit models all single-epoch catalogs in each area of sky together.

For the astrometric calibration, we will utilize a joint fit similar to that of Bernstein et al. (2017a) to constrain the instrumental degrees of freedom as well as a simple empirical (e.g. polynomial) model for astrometric offsets introduced by the atmosphere. This fit will be constrained to exactly reproduce the positions of stars measured by Gaia, which obtains much better astrometric accuracy on bright stars than LSST can hope to achieve; our goal here is essentially to extend the astrometric calibration to smaller spatial scales generally unconstrained by the shallower and hence sparser Gaia catalog.

The details of LSST's photometric calibration are still somewhat uncertain, depending in part on the ultimate photometric precision of the Gaia catalog. Joint modeling of multiple observations of the same objects will play a major role, just as in astrometric calibration, and we are currently integrating the forward-modeling approach used to calibrate the Dark Energy Survey (Burke et al. 2018) into our pipelines to extend this to physically-motivated modeling of the full photometric system. In the future, this will also incorporate the collimated beam projector data and auxiliary telescope spectra mentioned previously.

After joint calibration, we will return to single-epoch processing, however, to fit an improved PSF model by utilizing the more precise astrometric calibration and more secure star/galaxy classifications provided by the joint analysis of multiple epochs.

4.2. Coaddition and Image Differencing

At this point, our single-epoch images are almost fully characterized, but we expect to be able to improve on both our sky background model and our understanding of which pixels are affected by artifacts (e.g. cosmic rays, satellite trails, and optical glints and ghosts) via an image differencing analysis similar (but not identical) to that run in the Prompt pipelines.

This is particularly clear for artifact masking, because most artifacts will only appear in a single epoch, and even optical ghosts due to bright stars will appear in different positions as long as observations are dithered. The argument for background modeling is more subtle; it hinges on the fact that one of the biggest challenges in background estimation is separating variation in the sky from “astrophysical” backgrounds that we would like to preserve, such as diffuse light from galaxy clusters. Very few of these astrophysical backgrounds are time-variable, however, so they should subtract cleanly in image subtraction. Given N single-epoch images that cover some area of sky, then, we can construct difference image pairs that fully constrain $N - 1$ sky backgrounds. That leaves one “reference” image with its sky background intact and confused with the (common) astrophysical background. By combining the sky-subtracted images with the reference image to build a deeper coadd, we can enhance the signal-to-noise of the astrophysical background relative to the single remaining sky background, making them easier to separate in the final subtraction.

As we build coadds (including templates) and use them in image differencing analysis, we consequently also improve our background models and artifact masks. Just like the steps in initial characterization, the detailed ordering of the subsequent few processing steps is still to be determined due to circular dependencies that we must iteratively “unroll”. These steps include interpolating masked pixels, resampling all single-epoch images in each patch of sky to a common pixel grid, and comparing the resampled images to improve artifact masks and constrain sky-background pairs.

Once this iterative processing is complete, we can build final coadds and subtract the final background. This includes constructing template coadds to be used in image differencing in both the current Data Release and future Prompt processing; at this point we run essentially the same code in both productions, at least through DIASource production.

We also now return to single-epoch direct-image processing to produce the *Source* table, by performing one more round of detection, deblending, and measurement on the now fully-characterized single-epoch images.

4.3. Object Detection and Measurement

Entries in the *Object* table represent our best measurements of each astrophysical object in a given Data Release, and as a whole the table serves as a hub for other table data products. Objects can be produced in two ways: from coadd detections and DIAObjects (*i.e.*, DIASource associations). Sources derived from *direct* single-epoch detections are *not* directly associated into Objects; any astrophysical object should already appear in either coadd detections or DIASources (and be better characterized in at least one of those).

Detection on coadds will use the same algorithms employed on single-epoch direct (producing Sources) and difference images (producing DIASources), with the added complication that here we need consistent detections across all bands. We can achieve this either by combining coadds from different bands prior to detection (as in Szalay et al. 1999), or by detecting on the coadd for each band separately and merging them in catalog-space. We expect to use both approaches at some level (*i.e.*, detecting on different combinations of bands and merging the results). We may also create and detect on coadds created from inputs restricted to limited observation date ranges, in order to improve our ability to detect slowly-moving faint objects.

In Prompt processing, new DIASources are associated with and used to update living DIAObject and SSOBJect tables. Because we start from scratch in each Data Release, there we instead associate all DIASources in each patch of sky to immediately create complete DIAObjects and SSOBJects. DIAObjects are intended to represent astrophysical objects that either do not move or move very little, and hence each DIAObject is either associated with an existing Object derived from a coadd detection or used to create a new Object record. Because SSOBJects appear in a very different place every time they are observed, they are not included in the Object table. Whenever possible, we hope to have different Objects to reflect astrophysically distinct but spatially coincident entities, including distinguishing galaxy Objects from any time-variable AGN or supernovae they host, by utilizing time-domain information as well as positional information in the association.

Constructing secure Object definitions is probably impossible without adding morphological information into the mix, and this blurs the line we have drawn between detection and deblending in both the previous steps and in the current implementation of the pipeline. The Object deblending algorithm is very much still a work in progress, but the Scarlet algorithm (Melchior et al. 2018) is a likely starting point, and we can say that the full algorithm will utilize coadds from all bands simultaneously as well as DIAObject information, and it may modify the preliminary Object definitions passed to it from detection and association.

As in all deblending contexts, the Object deblender is also responsible for creating deblended child images. These are then used for measurement on coadds in approxi-

mately the same manner as in single-epoch processing, with the main differences being the set of algorithms that are run and the fact that at least some of these will utilize images from multiple bands simultaneously.

Two measurement algorithms will go considerably further, and fit models simultaneously to all of the original single-epoch images that overlap the Object. The first of these is a stellar astrometry model that fits a point source with proper motion and parallax parameters, which cannot be fit to coadds at all. Compared to traditional astrometry methods that fit motion parameters to independently-measured source positions, this approach *should* enable us to extend measurements to fainter magnitudes and better handle blending with faint neighbors (Lang et al. 2009).

The second model is a two-component PSF-convolved galaxy model, intended to approximate a bulge-disk decomposition (a full bulge-disk decomposition would require more signal-to-noise and resolution than the vast majority of LSST galaxies will have). At some level, this model could be fit only to the coadds, which should generally provide a good representation of the static sky. Utilizing the single-epoch pixel data makes it easier to guarantee that we have maximized the signal-to-noise of the measurement and avoided systematic errors (but only if all relevant single-epoch images are included in each likelihood evaluation). We are investigating approaches to building coadds of sufficient quality that going back to single-epoch images at this stage would be unnecessary, which would be much more computationally efficient. Systematic errors are the biggest concern, particularly those related to the print-through of CCD edges onto the coadd and noise correlations due to resampling. Methods exist to avoid signal-to-noise loss (e.g. Homrighausen et al. 2011; Zackay & Ofek 2017) in coaddition, but even without these the typical losses (which depend on the seeing distribution) seem to be minimal (Bosch et al. 2018).

Finally, we will return to both single-epoch direct images and difference images to perform forced photometry at the position of each Object. This should provide our best estimates of the light-curves of Objects, including Objects that were never detected in single-epoch images. We are currently planning to run this on both direct and difference images because direct images may provide slightly better signal-to-noise while difference images should provide much better control over systematics due to deblending. Deblending in single-epoch direct forced photometry – in which most blend children are not even detectable – is at some level impossible, and we generally expect difference-image forced photometry to provide better results overall, simply because blends should be much rarer and much less severe in difference images.

5. Conclusion

The LSST pipelines today are already representative of the state of the art in large-scale optical image processing. The future pipelines described here go considerably beyond this both in scale and in algorithmic sophistication, and there is hence substantial work to be done and a very good chance that some our of approaches to these problems will change before LSST first light. These challenges are present in both Prompt and Data Release processing, as well as in integrating both of these with each other and with the rest of the operational system.

Pushing the boundaries in software as well as hardware is a matter-of-course for new scientific projects, of course, and good progress is certainly being made. When

operational, the LSST pipelines will produce catalogs that enable similarly boundary-pushing science, in most cases without requiring further access to the pixel data.

Acknowledgments. We thank Keith Bechtol for helpful comments on earlier drafts of this paper. This material is based upon work supported in part by the National Science Foundation through Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.

References

- Aihara, H., et al. 2018, *PASJ*, 70, S4
- Alard, C., & Lupton, R. H. 1998, *ApJ*, 503, 325
- Barden, M., Häußler, B., Peng, C. Y., McIntosh, D. H., & Guo, Y. 2012, *MNRAS*, 422, 449
- Bernstein, G. M., Armstrong, R., Plazas, A. A., Walker, A. R., et al. 2017a, *PASP*, 129, 074503
- Bernstein, G. M., et al. 2017b, *PASP*, 129, 114502
- Bosch, J., et al. 2018, *PASJ*, 70, S5
- Brewer, B. J., Foreman-Mackey, D., & Hogg, D. W. 2013, *AJ*, 146, 7
- Burke, D. L., Rykoff, E. S., et al. 2018, *AJ*, 155, 41
- Coughlin, M., et al. 2016, in *Observatory Operations: Strategies, Processes, and Systems VI*, vol. 9910 of *Proc. SPIE*, 99100V
- Drlica-Wagner, A., et al. 2018, *The Astrophysical Journal Supplement Series*, 235, 33
- Gaia Collaboration, et al. 2016, *A&A*, 595, A1
- Homrighausen, D., Genovese, C. R., Connolly, A. J., Becker, A. C., & Owen, R. 2011, *Publications of the Astronomical Society of the Pacific*, 123, 1117
- Ivezić, Ž., & LSST Collaboration 2008, arXiv:0805.2366
- Jenness, T., et al. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of *ASP Conf. Ser.*, 653
- Jurić, M., et al. 2017, in *ADASS XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of *ASP Conf. Ser.*, 279
- 2018, *LSST Data Products Definition Document (LSE-163)*, <https://lse-163.lsst.io/>
- Lang, D., Hogg, D. W., Jester, S., & Rix, H.-W. 2009, *AJ*, 137, 4400
- Lupton, R., Gunn, J. E., Ivezić, Z., Knapp, G. R., & Kent, S. 2001, in *ADASS X*, edited by F. R. Harnden, Jr., F. A. Primini, & H. E. Payne, vol. 238 of *ASP Conf. Ser.*, 269
- Melchior, P., Moolekamp, F., Jerdee, M., Armstrong, R., Sun, A. L., Bosch, J., & Lupton, R. 2018, *Astronomy and Computing*, 24, 129
- O’Mullane, W., & LSST Data Management Team 2018, vol. 231 of *AAS Meeting Abstracts*, 362.10
- Regier, J., Pamnany, K., Giordano, R., Thomas, R., Schlegel, D., McAuliffe, J., & Prabhat 2016, arXiv:1611.03404
- Reiss, D. J., & Lupton, R. H. 2016, *Implementation of Image Difference Decorrelation (DMTN-021)*, <https://dmtn-021.lsst.io/>. doi:10.5281/zenodo.192833
- Schneider, M. D., Hogg, D. W., Marshall, P. J., Dawson, W. A., Meyers, J., Bard, D. J., & Lang, D. 2015, *ApJ*, 807, 87
- Sullivan, I. 2018, *DCR-matched template generation (DMTN-037)*, <https://dmtn-037.lsst.io/>. doi:10.5281/zenodo.1492936
- Swinbank, J., et al. 2017, *LSST DM Science Pipelines Design Document (LDM-151)*, <https://ldm-151.lsst.io/>
- Szalay, A. S., Connolly, A. J., & Szokoly, G. P. 1999, *AJ*, 117, 68
- Waters, C. Z., et al. 2016, arXiv:1612.05245
- Zackay, B., & Ofek, E. O. 2017, *ApJ*, 836, 188
- Zackay, B., Ofek, E. O., & Gal-Yam, A. 2016, *ApJ*, 830, 27

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Performance-related Aspects in the Big Data Astronomy Era: Architects in Software Optimization

Daniele Tavagnacco¹, Marco Frailis¹, Samuele Galeotta¹, Erik Romelli¹,
Davide Maino^{2,3}, Claudio Vuerli¹, Gianmarco Maggio¹, and Giuliano Taffoni¹

¹*INAF, Osservatorio Astronomico di Trieste, Trieste, Italy*

²*Università degli Studi di Milano, Milano, Italy*

³*INFN, National Institute for Nuclear Physics, Milano, Italy*

Abstract. In the last decades the amount of data collected by astronomical instruments and the evolution of computational demands have grown exponentially. Today it is not possible to obtain scientific results without prodigious amounts of computation. For this reason, the software performance plays a key role in modern Astronomy data analysis. Scientists tend to write code with the only goal of implementing the algorithm in order to achieve a solution: code modifications to gain better performance always come later. However, to facilitate this task, programming languages are progressing and introducing new features to fully make use of the hardware architecture. Designing a software that meets performance, memory efficiency, maintainability, and scalability requirements is a complex task that should be addressed by a software architect. In this paper we present the software refactoring and optimization activity performed at the Italian Science Data Center for the ESA's cosmological space mission Euclid.

1. Introduction

Euclid is a medium class astrophysics space mission of the *European Space Agency* (ESA) *Cosmic Vision* 2015-2025 scientific program. The satellite is a visible to near-infrared space telescope currently under development by the ESA and the *Euclid Consortium* (EC). The mission aims at exploring how the Universe evolved over the past 10 billion years in order to better understand the nature of the dark matter and the dark energy by measuring the modification of shapes of galaxies induced by weak gravitational lensing effects and the 3-dimension distribution of structures from spectroscopic redshifts of galaxies clustering. The satellite will be launched in 2022 and in a 6 years mission it will observe $\sim 15,000 \text{ deg}^2$ of sky. The final survey will consist in several thousands of images generating 30 PetaBytes of data (Euclid Consortium 2018). About 10 billion sources will be observed among which several million galaxy redshifts used for Galaxy Clustering (GC) measurements. The scientific data analysis and interpretation is led by the scientists of the EC formed by more than 1200 people in over 100 laboratories distributed in 15 countries. The huge volume, diversity and complexity of the data and the precision of the measurements to be performed, require a considerable effort in the data processing, making it a critical part of the mission. For this reason the EC is setting up top level teams of researchers and engineers in algorithm development, software development, testing and validation procedures, data archiving and data

distribution infrastructures for the Science Ground Segment (SGS) (Euclid Red Book Editorial Team 2011).

2. The Science Ground Segment

The Euclid ground segment consists of two blocks: the Operations Ground Segment and the Science Ground Segment (SGS) responsible for data processing and archiving. Due to the large amount of data that the mission will produce, the SGS is organized following a data-centric approach: all operations revolve around a central storage and inventory of the data products and their metadata. The large and reliable computing resources needed to process the whole data are provided to the SGS by the nine Science Data Centres (SDCs) located in participating countries of the EC and responsible of running different Processing Functions (PF) composing the data analysis pipeline. Any pipeline element is developed within the EC according to the following scheme:

- Requirements - Teams of scientists built around the Euclid science objectives called Science Working Groups (SWGs) define scientific requirements and the tests to check the validity of the pipeline element.
- Prototyping - The Organization Units (OUs) composed by EC scientists with code-development know-how perform algorithmic research by designing and developing software prototypes to satisfy SWGs requirements. There are no formal restrictions on the choice of infrastructure and language to be used.
- Integration - Once validated by SWG and OU, software prototypes are passed to their primary and their deputy SDCs to be turned into a production-ready Euclid pipeline element, which abides common coding standards, and uses pre-defined input and output mechanisms.

As the integration activity is divided into SDCs residing in different countries, a common software developing environment and a set of common tools for test and validation is needed. The homogenization of the code is consists in *porting* the prototype software into the two selected languages of the Euclid pipeline: C++ and Python. To facilitate and coordinate this activity, a common Euclid Development ENvironment (EDEN) has been defined. It consists in a set of C++, Python and external libraries and a set of *coding rules* defined by the EC. The Italian Science Data Center (SDC-IT) is the deputy SDC for the integration of the (Level 3) scientific processing functions. In particular SDC-IT is responsible for the galaxy clustering and clusters of galaxy software prototypes integration.

3. Code refactoring in SDC-IT

The Level 3 software is the most computational demanding part of the entire Euclid pipeline. This is mainly due to the need of reducing all the mission data into a scientific result humanly understandable. For this reason the integration of the Level 3 software prototypes in the Euclid pipeline requires also optimization. The process of restructuring an existing code without changing its external behavior is called refactoring. For commercial software usually refactoring is aimed to increase code maintainability, readability and extensibility and it is triggered mainly from *code smells*. In the case of

scientific software, refactoring happens to be an activity tightly coupled with the optimization and design part of the code life cycle. Indeed, scientific software development model remains highly waterfall-oriented (Ian Sommerville 2011) and, in many cases, the requirements are nor well defined or completely analyzed prior to the prototyping phase. The design phase is almost nonexistent since the development is mainly driven by the verification and testing phase where the algorithms are improved or even totally changed. This normally results in a huge, monolithic and multipurpose software that is highly inefficient and difficult to maintain. In these cases, software refactoring that includes a proper software design and optimization (E.Gamma, R.Helm, R.Johnson, J.Vlissides 1994) can significantly improve the code performances. The results of the refactoring activity performed by SDC-IT on a Level 3 GC processing function, namely the two point correlation function, that is aimed to measure the GC in excess or in defect compared to a random Poisson point distribution in space, is shown in Figs.1 and 2.

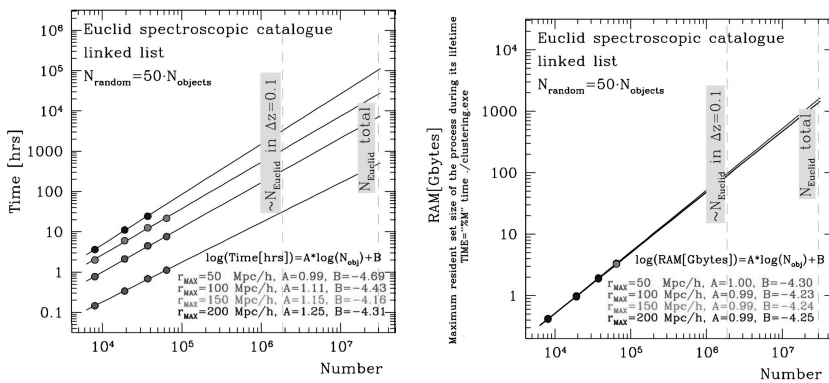


Figure 1. Two Point Correlation Function initial tests for code performance. The expected computing time (left) and memory usage (right) for typical Euclid GC catalog with 10^7 objects on a correlation scale of 200 Mpc/h for a spectroscopic galaxy sample can be only estimated since it requires too many resources.

In the example PF the original prototype code was developed in C and was a serial code. When selected for integration, some performance measurements were performed on it. Within the SDC-IT it has been ported to C++ and its design has been changed to obey to the Single responsibility, Open-close, Liskov's substitution, Interface segregation and Dependency inversion (SOLID) design principles. The main bottlenecks parts of the code have been substituted with simple code parallelization techniques. In this way the code is still maintainable by the original developers that are not software engineers. Most of the code parts have been replaced by proper C++ language features and the old *by-the-book* optimized sections have been replaced by new language features. This activity reduced the memory usage by 90% (from ~ 500 GB estimated to ~ 48 GB measured) and gained a speed up of $\sim 40\times$ in computing time (from 4×10^4 h estimated to ~ 500 h measured). Tests have been performed on a nominal Euclid-size catalog expected for GC computations containing $\sim 10^7$ galaxies considering a maximum separation between objects of 200 Mpc/h, typically used for BAO measurements (Daniel J. Eisenstein et al. 2005).

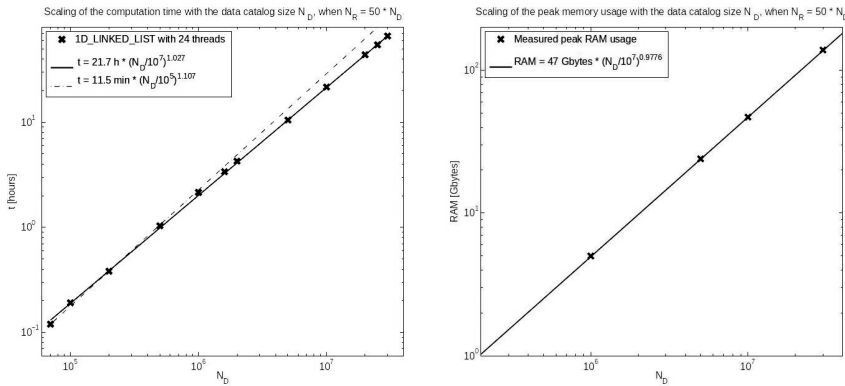


Figure 2. Two Point Correlation Function required resources as measured in 2017 code performance assessment. Computing time (left) and memory usage (right) for typical Euclid GC catalog with 10^7 objects on a correlation scale of 200 Mpc/h for a spectroscopic galaxy sample. Image from (Euclid Consortium 2017).

4. Conclusions

Scientific software is mainly developed by scientists with code-development know-how. It has a peculiar life cycle that requires a refactoring phase as a fundamental step for its optimization. In the prototype software we worked on for the integration in the Euclid pipeline we noticed that a huge improvement of code performances can be gained by properly following the coding rules and adopting the new optimized features provided by language evolution without relying on old optimization techniques already included in modern compilers. Moreover, in order to obtain a *big-data-ready* scientific software, in many cases it seems that keeping in mind performance when properly designing the code and when picking algorithms is preferable to the usual *make it work first, optimize later* technique that commonly works well for commercial software.

Acknowledgments. The authors acknowledge the Italian Space Agency (ASI), which supports the participation in Euclid of Italian institutions under grant no. I/023/12/0.

References

- Daniel J. Eisenstein et al. 2005, The Astrophysical Journal, 633, 560. arXiv:astro-ph/0501171
- E.Gamma, R.Helm, R.Johnson, J.Vlissides 1994, Design Patterns: Elements of Reusable Object-Oriented Software, 1st ed.
- Euclid Consortium 2017, Euclid SGS LE3 Software Design Document, Tech. rep.
- 2018, Euclid scientific ground segment data processing technical budget, Tech. rep.
- Euclid Red Book Editorial Team 2011, Euclid-Definition Study Report, 1st ed.
- Ian Sommerville 2011, Software Engineering, 9th ed.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

GWCS - A General Approach to Astronomical World Coordinates

Nadia Dencheva and Perry Greenfield

Space Telescope Science Institute, Baltimore, MD, USA; dencheva@stsci.edu

Abstract. GWCS is a package for managing the World Coordinate System (WCS) of astronomical data in a general way. We present the overall data model, new library developments, and tools for serializing the WCS object.

1. Introduction

The problem of expressing transformations from pixel to a standard coordinate system is fundamental to astronomy. The FITS WCS standard laid the foundation of the work in this area. As new instrumentation becomes available it is clear that the original FITS WCS model does not satisfy the current needs (Thomas et al. 2015) and a new paradigm is necessary.

The FITS WCS has also failed to deal with distortions well, making it impossible to use for precise imaging observations. This is generally a recognized problem and software packages which diverge from the FITS WCS standard already exist, for example, *AST* (Berry et al. 2016) and *astropy.wcs*. In addition there are many ad hoc solutions for complex data, particularly for spectra. We describe here the Generalized World Coordinate System (GWCS) library, designed to manage the WCS of astronomical data in a flexible way. While it supports the FITS WCS standard, its data model allows for much more general WCS representations.

2. The GWCS Library

2.1. Motivation

GWCS is an *astropy*-affiliated package written in Python and tightly integrated with *astropy* (Astropy Collaboration & Astropy Contributors 2018). It solves a number of issues we encountered while developing the WCS representations of Hubble Space Telescope (HST) images. The FITS WCS standard has no specification for representing distortions other than a few specific types of polynomials. A proposal for a more general handling of distortions in FITS files was never accepted. It had its own limitations - it did not allow distortions to be concatenated and it worked around the limitation of 8 character FITS keywords by devising a complex extension to the definition of a FITS keyword which was also not accepted. With Integral Field Unit (IFU) and Multiobject Spectrograph (MOS) observations becoming more popular it is clear that the FITS WCS standard does not meet their needs. GWCS is capable of representing the WCS of imaging, slit, IFU, MOS and grism observations. It is the basis of the James Webb

Space Telescope (JWST) WCS development. Nevertheless, it is written in a general way and is not specific to any mission.

2.2. The Data Model

In the context of GWCS the term “WCS” encapsulates the entire transformation pipeline, from an input coordinate frame to a standard celestial frame or a physical system. The data model is represented as a linear pipeline where steps are executed in sequence. Each step contains a coordinate frame and the transform to the next frame. The last step has a sentinel instead of a transform, designating the end of the WCS pipeline. The data model supports a WCS pipeline with intermediate frames. The python snippet below is an example of a typical WCS pipeline of an imaging observation.

```
>>> pipeline = [(detector_frame, distortion),
...             (undistorted_frame, det2sky),
...             (sky_frame, None)
...            ]

>>> wcsobj = wcs.WCS(pipeline)
>>> print(wcsobj)
From          Transform
-----
detector      distortion
undistorted_frame linear_transform
icrs          None
```

Transforms in GWCS are instances of *astropy.Model* and use the flexible framework for model combination in *astropy.modeling*. In general they map “n” inputs into “m” outputs, using if necessary special models which help with dimension management (axes can be swapped, dropped or added). Transforms can be chained (combined in series), joined (combined in parallel) or combined using arithmetic functions. Where appropriate transforms support units. They can be initialized and evaluated with quantities (numbers with units attached to them) allowing for automatic unit conversion during evaluation and fitting. Many analytical models and their inverses are already defined in *astropy.modeling* and the framework allows for easy addition of new models and operators. In addition, GWCS provides some WCS specific transforms, specifically those dealing with discontinuous WCSs (e.g. IFU). These transforms essentially provide a mapping between pixels and slice labels and pixels and slice specific transforms.

Coordinate frames are mostly informational containers with attributes such as *axes_names*, *axes_type*, *axes_order*, *axes_physical_type*, *unit*. Each coordinate frame provides a method that converts the numerical results to corresponding *astropy* objects capable of handling transformations using the tools in *astropy*. Celestial coordinates are instances of *astropy.SkyCoord* and can be further transformed to other standard celestial frames available in *astropy.coordinates*. Time coordinates are instances of *astropy.Time*. They can be manipulated using the tools in *astropy.time* which deal with time scales and time representations. Spectral coordinates are *astropy.Quantity* objects and can be converted to other spectral units with the tools in the *astropy.units* package, using equivalencies if necessary. GWCS allows also for the specification of custom frames.

2.3. WCS Tools

GWCS implements the *Shared WCS API* defined in Astropy Proposal for Enhancements 14 (APE14, Robitaille et al. 2018). The API defines a standardized interface to WCS objects based on strings, scalars and arrays such that packages that use the WCS do not need to understand the underlying object. The API aims to improve interoperability between packages by setting a common understanding of how to convey information about the physical type and representation of a world coordinate in Python. The Shared API exposes the two most often used methods, *pixel_to_world* and *world_to_pixel*, and provides access to the underlying WCS implementation.

GWCS provides also methods for transforming coordinates between any two frames in the WCS pipeline. Transforms between frames can be manipulated or completely replaced, thus facilitating common tasks like aligning images using their WCSs.

A new tool in GWCS takes two matching sets of points, representing measured coordinates on the detector and the corresponding known sky coordinates, and returns a WCS object.

3. WCS Serialization

GWCS utilizes the Advanced Scientific Data Format (ASDF, Greenfield et al. 2015) to serialize and validate GWCS objects. A WCS can be saved either to an ASDF file or as an ASDF extension to a FITS file. ASDF makes use of abstract data types called *schemas*. Serialization happens in classes referred to as *tags* which convert to and from astropy objects. ASDF files are YAML files with numerical arrays saved as binary blocks. ASDF uses jsonschema to validate the structure of the objects and their metadata. The validation happens transparently to the end user making it unnecessary to know the details of schemas and YAML. However, packages using GWCS may create their own transforms and schemas and make them available by registering them as an ASDF extension.

3.1. Example saving a WCS object to a pure ASDF file

```
>>> from asdf import AsdfFile
>>> tree = {"wcs": wcsobj}
>>> wcs_file = AsdfFile(tree)
>>> wcs_file.write_to("imaging_wcs.asdf")
```

3.2. Example saving a WCS object to an ASDF extension in a FITS file

```
>>> from astropy.io import fits
>>> from asdf import fits_embed
>>> hdul = fits.open("example_imaging.fits")

>>> hdul.info()
Filename: example_imaging.fits
No.      Name      Ver  Type      Cards  Dimensions  Format
0  PRIMARY          1 PrimaryHDU    775      ()
1  SCI              1 ImageHDU     71    (600, 550)  float32
```



```
>>> tree = {"sci", hdul.data,
...         "wcs": wcsobj}

>>> fa = fits.embed.AsdfInFits(hdul, tree)

>>> fa.write_to("imaging_with_wcs_in_asdf.fits")

>>> fits.info("imaging_with_wcs_in_asdf.fits")
Filename: example_with_wcs.asdf
No.    Name      Ver   Type      Cards  Dimensions  Format
0  PRIMARY      1  PrimaryHDU    775    ()
1  SCI          1  ImageHDU     71    (600, 550)  float32
2  ASDF         1  BinTableHDU   11    1R x 1C    [5086B]
```

References

- Astropy Collaboration, & Astropy Contributors 2018, AJ, 156, 123
- Berry, D., Warren-Smith, R., & Jenness, T. 2016, Astronomy and Computing, 15, 33
- Greenfield, P., Droettboom, M., & Bray, E. 2015, Astronomy and Computing, 12, 240
- Robitaille, T., Tollerud, E., Mumford, S., & Ginsburg, A. 2018, Astropy Proposal for Enhancement 14: A shared Python interface for World Coordinate Systems (APE 14). URL <https://zenodo.org/record/1188875#.XBGDn5x0lHc>
- Thomas, B., et al. 2015, Astronomy and Computing, 12, 133. arXiv:1502.00996



Poster boards in the back of the lecture hall. (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Data-Driven Pixelation with Voronoi Tessellation

Marco C. Lam and Paul R. McWhirter

*Astrophysics Research Institute, Liverpool John Moores University, IC2,
 Liverpool Science Park, 146 Brownlow Hill, Liverpool L3 5RF, UK;
 C.Y.Lam@ljmu.ac.uk*

Abstract. In modern Astrophysics, Voronoi Tessellation is a rarely used as a pixelation scheme. While it exists, it is almost exclusively used in signal enhancements, clustering analysis, and simulations. In observational Astronomy, with Gaia, ZTF, DES, etc. data becoming available, this branch of science is becoming more and more data-driven. HEALPix offers excellent ways to pixelize the celestial sphere, the implementations completely separate the background information from the data. While with Voronoi Tessellation, it can generate a one-to-one mapping of data points to Voronoi cells. This concept also works on a HEALPix scheme, which enable users with minimal knowledge on spherical trigonometry to apply this method. We discuss how this can be used to detrend light curves data from the Liverpool Telescope SkyCam-T.

1. Introduction

The study of vast number of transient, variable, and moving sources in Astronomy has been enabled by the advance in astronomical instrumentation and the increase in computing power. The highly automated observing runs and efficient digital detectors allow efficient data collection, while faster processors and the automated data reduction pipelines allow the production of high volume of output. However, the complex observing strategies and small scale variations over the detector plane have made the survey properties extremely difficult to model. This in turn complicates the analysis of samples drawn from such kind of multi-epoch large sky area surveys, particularly the completeness function at different parts of the sky.

If an averaged global survey limits is applied to an analysis, we sacrifice some sources with good quality as the selection function is skewed by the lower quality data. Hence, in order to maximize the yield from a survey, one should avoid global approximations and use the local information where possible. For example, Lam (2017) demonstrated how the propagation of instrumental noise into photometric and astrometric uncertainties has enabled the use of individual uncertainties in constructing a luminosity function. The simulations also show that it can recover 25% more sources by switching from a global to a local approach to model the proper motion uncertainties as a function of magnitude¹ – fitting a spline through the 95th percentile. It was motivated by the tessellation of the sky with photographic plates (Rowell & Hambly 2011).

¹The function was characterized with $5^\circ \times 5^\circ$ tiles.

When working with a source catalogue with limited information of the survey properties, partitioning the survey area by a random pixelation scheme generates areas with no sources. This raises the question of how one can get the properties of these empty areas, when we are only supplied with a catalogue of processed data, for example, the Gaia DR2 (Gaia Collaboration et al. 2018).

2. Pixelation Schemes

One of the most commonly used pixelation schemes in Astronomy is the Hierarchical Equal Area isoLatitude Pixelation – HEALPix² (Górski et al. 2005); it divides the sky into 4-sided pixels with equal area, starting at 12 pixels at the lowest resolution level. Each successive level of resolution sub-divides the pixel into 4 equal area polygons. It has very efficient and has a huge number of use cases, but data-driven science is not one of those. Voronoi Tessellation is made by partitioning a plane with n points into n convex polygons. Any position in a given Voronoi cell is closer to its generating point than to any of the neighbours. The loci equal-distance from the data points trace the boundaries of the cells. To construct a Voronoi Tessellation on a sphere, it can be done by applying a 3D convex hull to the data, which is equivalent to applying Delaunay Triangulation. The circumcenters of all triangles formed from the edges give the vertices of the Voronoi cells. This work uses the Scipy package `spatial.SphericalVoronoi`. The area of a cell can be calculated from the sum of the spherical excess of the constituent triangles, which can be found from the L'Huilier's Theorem.

3. Footprint Area and Reproducibility

A Voronoi tessellation on a closed surface is always bound. However, it is always constructed over the entire sphere with a total solid angle of 4π . Among Astronomical surveys, it is rare to have full sky coverage. In the case of luminosity function, we need to know the area in order to get the survey volume. One common case is that in order to avoid crowding issues, a band 10° from the Galactic plane and the 20° from the Galactic centre are often masked out. Thus, it is necessary to add artificial points in order to generate the footprint area of interest. The left panel of Figure 1 shows how such implementation works on a sphere, using the packages aforementioned.

This procedure is quick and easy for a survey with simple geometry. However, for most surveys, the footprint is not so easily defined and it often involves a lot of spherical transformation and trigonometry, which is not user-friendly. In such cases, it is possible to apply the concept of equal-distance properties to a grid of HEALPix pixels³ and map the nearest pixels to the generating data points. This can work nicely in conjunction with a footprint service (e.g., Budavári et al. 2007, 2010) to define the area for analysis. Unsurprisingly, at a sufficiently fine resolution, they can approximate the Voronoi cells very well (see the right panel of Figure 1). Instead of computing the

²<https://healpix.sourceforge.io/>

³It is suggested rather than HTM because of its equal area property so that the solid angle can be found by multiplying the solid angle at that resolution with the number of pixel in a selected region.

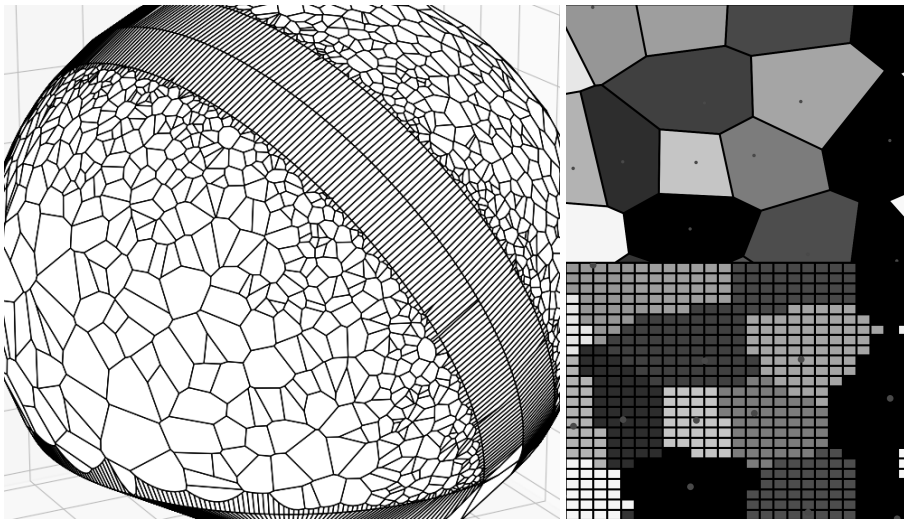


Figure 1. Left: Three pairs of rings of equally spaced artificial points were added along the small circles 10° from the Galactic Plane, and at Declination of -30° . Top Right: Voronoi tessellation of some random points on a 2D plane. Bottom Right: HEALPix at a high resolution that the pixels are smaller than the Voronoi cells.

properties of each HEALPix pixel, only the Voronoi cell-shaped “mega-pixels” have to be calculated, based on the properties of the generating data points.

The minimum requirement to reproduce the analysis is the noise properties of the set of the generating points and the footprint geometry. It was demonstrated to work effectively with a proper motion selection function, where epoch level information of each source is required to reproduce the analysis (Lam et al. 2019). The supplementary data in that work contains epoch from $\sim 15,000$ data points from the Pan-STARRS 1 3π Steradian Survey (Chambers et al. 2016), which is a massive contraction from modelling 10^5 pointings with 60 CCDs each with 64 regions⁴ over 3.5 years. When working in conjunction with another pixelation scheme, some extra meta-information (e.g., pixelation level/resolution) are also required.

4. Detrending Liverpool Telescope SkyCam-T Light-Curves

The SkyCam-T is a small telescope mounted on the LT top-end which parallel points with the telescope. It has a $\sim 9^\circ \times 9^\circ$ field-of-view. Some 600,000 light-curves were generated between 2009 and 2012 (Mawson et al. 2013). In an effort to classify variable stars with GRAPES (McWhirter et al. 2018), it was found that there was significant seasonal trending in the light-curves. It is proposed to apply the Voronoi Tessellation to the centers of the clustering of pointings (see Figure 2) with a large number of epochs and aggregate light-curves within the regions to model the trend (Kovács et al. 2005).

⁴A lot of chip defects and gaps!

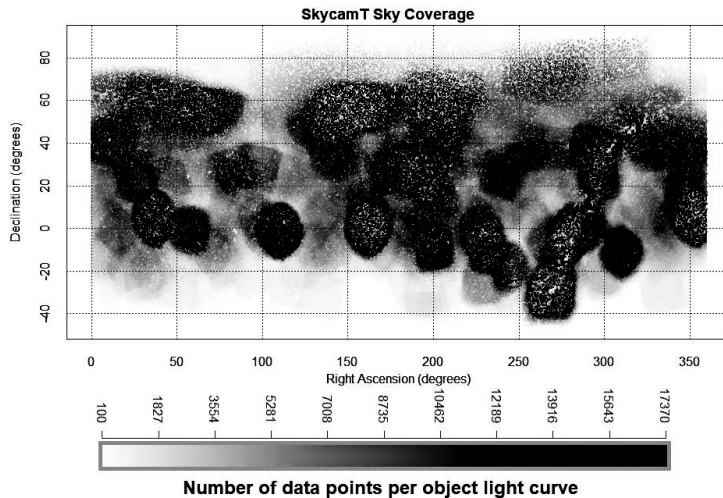


Figure 2. Number of data points per object in each light-curve on the sky from SkyCam-T in Cartesian projection.

5. Conclusion

The construction of a Voronoi tessellation only requires the data points. In order to reproduce the tessellation, an identical set of points is required. It is flexible that we need to know little about the survey and the data themselves can provide the necessary information. On the other hand, different samples drawn from the same survey can provide different survey properties. Using a hybrid approach to define the footprint with HEALPix and apply the concept of Voronoi tessellation can allow a simple way to maximize the analytical survey volume, without the need to work with spherical trigonometry.

References

- Budavári, T., Dobos, L., Szalay, A. S., Greene, G., Gray, J., & Rots, A. H. 2007, in *Astronomical Data Analysis Software and Systems XVI*, edited by R. A. Shaw, F. Hill, & D. J. Bell, vol. 376 of *Astronomical Society of the Pacific Conference Series*, 559
- Budavári, T., Szalay, A. S., & Fekete, G. 2010, *PASP*, 122, 1375. 1005.2606
- Chambers, K. C., et al. 2016, *ArXiv e-prints*. 1612.05560
- Gaia Collaboration, et al. 2018, *A&A*, 616, A1. 1804.09365
- Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. 2005, *ApJ*, 622, 759. astro-ph/0409513
- Kovács, G., Bakos, G., & Noyes, R. W. 2005, *MNRAS*, 356, 557. astro-ph/0411724
- Lam, M. C. 2017, *MNRAS*, 469, 1026. 1704.08745
- Lam, M. C., et al. 2019, *MNRAS*, 482, 715. 1810.01798
- Mawson, N. R., Steele, I. A., & Smith, R. J. 2013, *Astronomische Nachrichten*, 334, 729. 1305.0573
- McWhirter, P. R., Steele, I. A., Hussain, A., Al-Jumeily, D., & Vellasco, M. M. B. R. 2018, *MNRAS*, 479, 5196. 1807.07010
- Rowell, N., & Hambly, N. C. 2011, *MNRAS*, 417, 93

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

The JWST Data Calibration Pipeline

Howard Bushouse, Jonathan Eisenhamer, and James Davies

Space Telescope Science Institute, Baltimore, MD, USA; bushouse@stsci.edu

Abstract. STScI is developing the software systems that will provide routine calibration of the science data received from the James Webb Space Telescope (JWST). The processing uses an environment provided by a Python module called `stpipe` that provides many common services to each calibration step, relieving step developers from having to implement such functionality. The `stpipe` module provides common configuration handling, parameter validation and persistence, and I/O management. Individual steps are written as Python classes that can be invoked individually from within Python or from the `stpipe` command line. Any set of step classes can be configured into a pipeline, with `stpipe` handling the flow of data between steps. The `stpipe` environment includes the use of standard data models. The data models, defined using `yaml` schema, provide a means of validating the correct format of the data files presented to the pipeline, as well as presenting an abstract interface to isolate the calibration steps from details of how the data are stored on disk.

1. Introduction

STScI has developed and maintained data calibration pipelines for all of the HST scientific instruments and is now in the process of developing the pipelines that will be used for the James Webb Space Telescope (JWST). The HST pipelines were developed over a span of more than 20 years and hence show an evolution in both software languages and design. The pipelines for each instrument – a total of 11 over the history of HST — were written mostly independently of one another and used an assortment of languages, ranging from IRAF SPP to Fortran, C, and Python. This made maintenance and enhancement rather difficult, and precluded much code sharing between instruments. The HST pipelines also used monolithic, procedural designs, with very little modularity. This approach worked as long as data were allowed to flow uninterrupted from beginning to end, but made it very difficult, if not impossible, to start or stop processing midstream, skip one or more steps, or insert additional steps.

The JWST calibration pipelines are being developed using a completely new design approach using mostly Python. There is a common framework for all 5 of the scientific instruments, with extensive sharing of routines and a common code base. The new design allows for flexibility in swapping in and out specific processing steps, easily changing the ordering of steps within pipelines, and the ability for astronomers to insert custom routines. The calibration pipelines will be distributed to astronomers, giving them the ability to rerun and refine the processing of their observations. The highly modular and flexible nature of the design will allow them to add custom processing steps, either as part of the pipeline itself or as standalone routines that are run on the data and then reinserted back into the pipeline flow. The calibration pipeline package

has been designed to be as light-weight and self-contained as possible in order to make it easy for users to install and run. The only external interface required is to our Calibration Reference Data System (CRDS), which is used to supply reference data needed by the calibration steps. The CRDS server at STScI will accept requests for reference files from the client on an astronomer's home system and automatically download the requested files to their systems for use locally.

2. **stpipe**

The central nervous system of the JWST calibration pipeline environment is a Python module called **stpipe**. **stpipe** manages individual processing steps that can be combined into pipelines. The **stpipe** environment provides functionality that is common to all steps and pipelines so that they behave in a consistent manner. It provides for running steps and pipelines from the command line, parsing of configuration settings, composing steps into pipelines, file management and data I/O between pipeline steps, an interface to the CRDS, and logging.

Each step is embodied as a Python class, with a pipeline being composed of multiple steps. Pipelines can in turn be strung together, just like steps, to compose an even higher-order flow. Steps and pipelines can be executed from the command-line using **stpipe**, which is the normal mode of operations in the production environment. Step and pipeline classes can also be instantiated and executed from within a Python shell, which provides a lot of flexibility for developers when testing the code and to astronomers who may need to tweak or otherwise customize the processing.

When run from the command line, **stpipe** handles the parsing of configuration parameters, which can be provided either as arguments on the command line or within configuration files. Configuration files use the well-known ini-file format and **stpipe** uses the ConfigObj library to parse them. **stpipe** handles all of the file I/O for each step and the passing of data between pipeline steps, as well as providing access within each step to a common logging facility. It also provides a common interface for all steps to reference data files that are stored in the CRDS. Having all of these functions handled by the **stpipe** environment relieves developers from having to include these features in each step and provides a consistent interface to users as well.

stpipe is used to execute a step or pipeline by providing either the class name of the desired step/pipeline or a configuration file that references the step/pipeline class and provides optional argument values. An example that directly calls a class is:

```
> strun jwst.pipeline.SloperPipeline input.fits --output_file="myimage.fits"
```

The same thing can be accomplished by specifying a config file, e.g.:

```
> strun sloper.cfg input.fits
```

where sloper.cfg contains:

```
name = "SloperPipeline"
class = "jwst.pipeline.SloperPipeline"
output_file = "myimage.fits"
save_calibrated_ramp = True
```

Steps and pipelines can be called from Python using the class' "call" method:

```
>>> from jwst.pipeline import SloperPipeline
>>> result=SloperPipeline.call('input.fits', config_file='sloper.cfg')
```


The `stpipe` logging mechanism is based on the standard Python logging framework. The framework has certain built-in things that it automatically logs, such as the step and pipeline start/stop times, as well as platform information. Steps can log their own specific items and every log entry is time-stamped. Every log message that's posted has an associated level of severity, including `DEBUG`, `INFO`, `WARN`, `ERROR`, and `CRITICAL`. The user can control how verbose the logging is via arguments in the config file or on the command line.

3. Steps and Pipelines

Steps define parameters, their data types (in “`configspec`” format), and default values. As mentioned earlier, users can override the default parameter values by supplying values in configuration files or on the command-line. Steps can be combined into pipelines, and pipelines are themselves steps, allowing for arbitrary levels of nesting.

Simple linear pipelines can be constructed as a straight sequence of steps, where the output of each step feeds into the input of the next. These linear pipelines can be started and stopped at arbitrary points, via arguments supplied by the user, with all of the status saved to disk and then resumed later if desired. More complex (non-linear) pipelines can be defined using a Python function, so that the flow between steps is completely flexible. Because of their non-linear nature, these types of pipeline can not be started or stopped mid-stream. Both types of pipelines, however, allow the user to skip steps by supplying configuration overrides.

Step configuration files can also specify pre- and post-hooks, to introduce custom processing into the pipeline. The hooks can be Python functions or shell commands. This allows astronomers to examine or modify data, or insert a custom correction, at any point along the pipeline without needing to write their own Python code.

Excerpts (for brevity) of a pipeline are shown below. In this example, the input data is modified in-place by each processing step and the results passed along from one step to the next. The final result is saved to disk by the `stpipe` environment. Each pipeline subclass inherits from the `Pipeline` class. The subclass defines the Steps that will be used so that the framework can configure parameters for the individual Steps. This is done with the `step_defs` member, which is a dictionary that maps step names to step classes. This dictionary defines what the Steps are, but says nothing about their order or how data flows from one Step to the next. That is defined in Python code in the Pipeline's `process` method. By the time the Pipeline's `process` method is called, the Steps in `step_defs` will be instantiated as member variables.

```
from jwst.stpipe import Pipeline
from jwst.dq import dq_step
from jwst.ramp import ramp_step

# the pipeline class
class SloperPipeline(Pipeline)
# step definitions
step_defs = {"dq" : dq_step.DQInitStep,
             "ramp_fit" : ramp_step.RampFitStep}

# the pipeline process
def process(self, input):
    log.info("Starting calwebb_sloper ...")
    input = self.dq(input)
```



```

# only apply reset and lastframe to MIRI data
if input.meta.instrument.name == "MIRI":
    input = self.reset(input)
    input = self.frame(input)
    input = self.jump(input)
# save the results so far
if self.save_cal:
    self.save_model(input, "ramp")
    input = self.ramp_fit(input)
    log.info("... ending calwebb_sloper")
return input

```

4. Data Models

The burden of loading, parsing, and interpreting the contents of FITS data files usually falls to the processing code that's trying to do something with the data. For the JWST calibration pipelines, the `stpipe` environment takes care of all the file I/O, leaving the developers to concentrate on processing the data. This is accomplished through the use of software data models. The data models allow the on-disk representation of the data to be abstracted from the pipeline steps via the I/O mechanisms built into `stpipe`. The use of data models also has the benefit of eliminating or at least being able to manage dependencies between the various steps. Because all of the actual science data and its meta data are completely self-contained within a model, each step has all of the information it needs to do its work. If a particular processing step changes the overall format or content of the data in some way, the result is saved in a different type of data model. Each step can perform a check to ensure that the input it's been given conforms to the type of model expected in that step. Any inconsistencies will be detected immediately and the process will shutdown with a warning to the user, rather than the undesirable behavior of having a step crash because the input data were not compatible.

The models interface currently reads and writes FITS files, as well as the Advanced Scientific Data Format (ASDF) file format developed by STScI. The interface provides the same methods of access within the pipeline steps whether the data are on disk or already in memory. Furthermore, the interface can decide the best way to manage memory, rather than leaving it up to the step code. The use of data models also isolates the processing code from future changes in file formats or keywords. Each model is a bundle of array or tabular data, and meta data, with the model structure defined using schemas in YAML format. The model schemas are modular, such that a core schema that contains elements common to all models can include any number of additional sub-schema that are unique to one or more particular models.

Step code loads a data model using a simple statement like:

```
im = datamodels.ImageModel(input)
```

`stpipe` determines whether "input" is a model already in memory or a file on disk. If the latter, it loads the file into an `ImageModel`. The step code then has direct access to all attributes of the `ImageModel`, such as the data, dq, and error arrays defined in the `ImageModel` schema. If only a single step is executed, `stpipe` saves the returned data model to disk. If part of a pipeline, `stpipe` passes the returned data model in memory to the next step and saves the final result at the end of the pipeline.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Jitter and Readout Sampling Frequency Impact on the Athena/X-IFU Performance

M.T. Ceballos,¹ B. Cobo,¹ and P. Peille²

¹*Instituto de Física de Cantabria (CSIC-UC), Santander, Spain;*
ceballos@ifca.unican.es

²*CNES, Toulouse, France*

Abstract. The TES calorimeters like the one to be the core of the X-IFU instrument on board the Athena mission, provides unprecedented energy resolution. However, this resolution can be compromised due to the non-perfect sampling of the pulses rising-edges (jitter) which could result in significant errors of the reconstructed energies if the standard optimal filtering algorithm is applied (with no further correction). If not corrected properly, these errors could induce a prohibitive broadening of the energy resolution. We present here the analysis of the magnitude of this effect and propose a correction. In addition, we evaluate the impact of a reduced readout sampling frequency in the energy resolution, once the jitter correction has been applied.

1. Introduction

The X-ray Observatory Athena(Nandra et al. 2013) is the mission selected by ESA to implement the science theme The Hot and Energetic Universe for L2 (the second Large-class mission in ESA's Cosmic Vision science program). One of the two X-ray detectors designed to be onboard Athena is X-IFU (X-ray Integral Field Unit)(Barret et al. 2018), a cryogenic microcalorimeter based on Transition Edge Sensor (TES) technology that will provide spatially resolved high-resolution spectroscopy.

The X-ray photons absorbed by X-IFU detector generate intensity pulses that must be detected and reconstructed on-board to recover their energy, position and arrival time. During the detection stage, the single pulses that the data stream may contain are individually triggered and assigned a trigger time. During reconstruction, an initial estimation of the pulse height (as a proxy for the energy) of the pulses is performed. The baseline for the processing algorithms has been defined selecting the optimal filtering for energy reconstruction(Peille et al. 2016) and the Single Threshold Crossing(Cobo et al. 2018) as the triggering mechanism. This combined selection provides the best compromise results for the mission requirements, based on the analysis of the simulated data. However, due to the non-perfect sampling of the pulses rising edges (jitter), the random offset between the actual pulse arrival times and the trigger times could result in significant errors of the reconstructed pulse heights. If not corrected properly, these errors could induce a prohibitive broadening of the energy resolution.

An initial correction is done during the reconstruction process, where each event is optimally filtered on phase with the trigger time and with a ± 1 sample offset. This results in three pulse height (PH) estimates forming a parabola. Its peak is taken as

the “final” height estimate and the distance of the abscissa’s peak and the trigger time, as its arrival phase with respect to the sampling process(Adams et al. 2009). Ideally, this PH estimate would not need to be corrected, just calibrated through the gain scale function to get the energy values in the proper units. However, in some configurations this calibration would not be enough and the energy resolution could be compromised.

The main purpose of this study is the analysis of the situations where an additional correction would be required and of the effects of the proposed solution, in particular its dependence with the readout sampling rate, since the limitations of this correction would be one of the key elements for the choice of its final value.

2. Simulation framework

This study is based on simulations performed with the xifusim tool of the X-IFU end-to-end simulator (based on SIXTE/tessim(Wilms et al. 2016)) for LPA75um pixels, with the Base Band Feed Back (BBFB) control loop. Data have been simulated for three different sampling rates: 156250 Hz (S1), 78125 Hz (S2), 39062.5 Hz (S4).

The optimal filtering technique requires the construction of an optimal filter by means of a pulse template and a noise spectrum. The template has been built averaging 20000 high resolution (perfectly isolated) 6 keV pulses simulated at random offsets with the sampling (between -0.5 and +0.5). To convert pulse heights obtained with the reconstruction process into proper unit energies a gain scale (functional relation between the input simulated energy and the output reconstructed pulse height) must be derived. For this purpose 5000 high resolution (perfectly isolated) pulses have been simulated at calibration energies of 1, 2, 3, 4, 5, 6, 7 and 8 keV and with random offsets between -0.5 and 0.5. These pulses are also used in Sec3 to illustrate the resolution dependence with the arrival offset.

The testing of the proposed correction is performed over 2000 high resolution (perfectly isolated) simulated pulses with random offsets between -0.5 and 0.5. They are used to show the energy resolution achieved after the initial parabola correction, after the gain scale calibration and finally, after the proposed polynomial correction.

3. Resolution and arrival phase

The optimal filter reconstruction of the pulses shows a clear dependence with the arrival offset of the pulses, more pronounced for the lowest sampling rates. Fig.1 shows, for different energies, the relative remaining error of the gain-scale calibrated energy with respect to the input simulated energy, as a function of the measured arrival phase.

The measured arrival phase has a bias with respect to the true arrival phase, which depends on the energy. The pulse shape changing with energy causes the dependence of the measured phases on the energy. The effect is especially pronounced for lower sampling rates and thus needs to be corrected. In addition, the relation between the measured arrival phase and the estimation error depends strongly on the energy not being easily corrected with a single relation calibrated at a given energy.

In the figure 2 the effect over the reconstructed (and gain scale calibrated) energies can be seen for the 2000 simulated pulses at 7 keV for the 3 sampling rates included in this study. The plot suggests that for the largest sampling rate (156.25kHz) no addi-

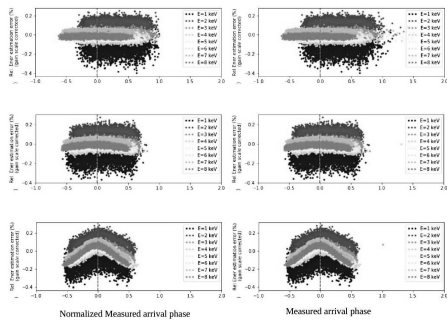


Figure 1. Relative Energy Estimation Error vs arrival Phase for sampling rates S1 (top), S2 (middle) and S4 (bottom). Phases on the right are as derived from the processing and those on the left have been normalized to (0,1) samples.

tional correction would be required to comply with the resolution requirements but for the other two lower values it would be mandatory.

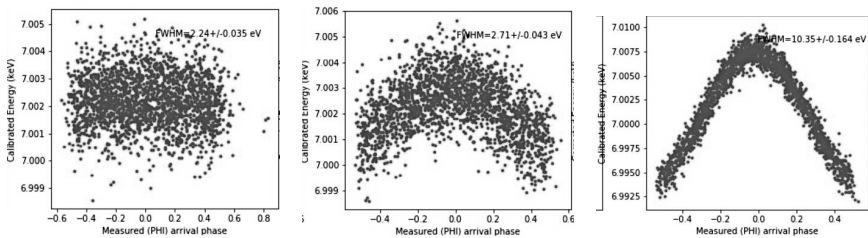


Figure 2. Reconstructed (gain-scale calibrated) energy vs measured arrival phase for sampling rates S1 (left), S2 (center) and S4 (right) and for 7 keV simulated pulses.

4. Calibration of the correction

The energy dependence of the jitter correction excludes the possibility of applying an initial single pulse jitter correction followed by the standard gain scale function. Another possibility would be applying the standard gain scale function first and then interpolate a pulse jitter correction from relations calibrated at different energies. This might however leave residuals due to the non-perfect interpolation of the jitter correction function. The approach used here is to apply directly a 2D gain scale function that depends on both the measured pulse height and arrival phase. This is in principle the most accurate correction but also the one that would be the most difficult to calibrate. For this purpose the 5000 simulated pulses at various phases and energies were used to fit a sixth-degree polynomial 2D gain function from their associate reconstructed pulse heights and arrival phases (input simulated energy is known). This way, when a pulse is reconstructed, instead of just measuring the “peak” energy and applying a gain scale as $f(PH)$, one needs to measure its phase too and apply a 2D gain scale as

$E = g(PH, phase)$. For the calibration, what is needed is to have a set of measured pulse heights and phases for different energies.

5. The sampling rate factor

As it was shown in Fig.2 a key factor for the magnitude of the arrival phase bias (and thus for its correction) is the sampling rate: the lower the sampling rate, the larger the jitter effect, and the more difficult it will be to calibrate the correction. For the evaluation of the influence of the sampling rate over the performance of the polynomial correction, the 2000 pulses simulated at different energies and sampling rates were reconstructed and corrected using the 2D gain-scale functions derived from the polynomial fits at every sampling rate. The results of the corrections are shown below.

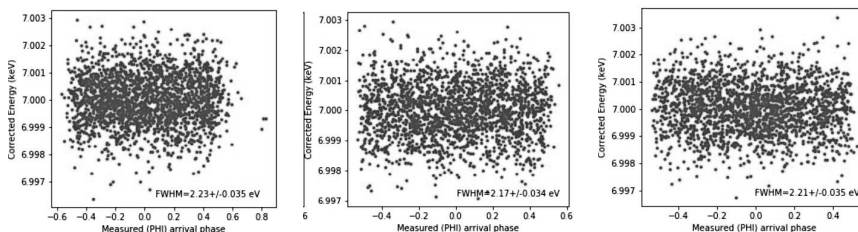


Figure 3. Reconstructed (2D-gain-scale corrected) energy vs measured arrival phase for sampling rates S1 (left), S2 (center) and S4 (right) and for 7 keV simulated pulses.

6. Conclusions

After this preliminary study, the main initial conclusions would be: (1) the jitter effect does not seem to be very relevant for the nominal 156.25kHz sampling rate but it causes a prohibitive degradation of the energy resolution at a two times lower (78.125 kHz) and four times lower (39.0625 kHz) sampling rates; (2) A 2D gain scale is an efficient way to obtain a calibrated energy estimate from the raw phase and pulse height to significantly remove the effects of the jitter bias, specially for the lowest readout sampling rates considered. After this correction there is no apparent degradation of the reconstruction performance.

Acknowledgments. This work has been funded by the Spanish Ministry MINECO under project ESP2016-76683-C3-1-R, co-funded by FEDER funds.

References

- Adams, J., et al. 2009, in LTD 13, vol. 1185 of AIP Conference Proceedings, 274
- Barret, D., et al. 2018, in SPIE 2018: Ultraviolet to Gamma Ray, vol. 10699 of SPIE Conf. Ser.
- Cobo, B., et al. 2018, in SPIE 2018: Ultraviolet to Gamma Ray, vol. 10699 of SPIE Conf. Ser.
- Nandra, K., et al. 2013, ArXiv e-prints. 1306.2307
- Peille, P., et al. 2016, in SPIE 2016: Ultraviolet to Gamma Ray, vol. 9905 of SPIE Conf. Series
- Wilms, J., et al. 2016, in SPIE 2016: Ultraviolet to Gamma Ray, vol. 9905 of SPIE Conf. Series

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

A Simple Survey of Cross-Matching Methods

Dongwei Fan, Yunfei Xu, Jun Han, Changhua Li, Boliang He, Yihan Tao, and
Chenzhou Cui

*National Astronomical Observatories, Chinese Academy of Sciences, Beijing,
China; fandongwei@nao.cas.cn*

Abstract. In order to find a practicable method to build an online cross-matching service, we test several index and search methods. Indexing methods include the directly matching, HEALPix based matching and zones algorithm. Sorted list and several search trees are also inspected, e.g., Binary Search Tree, Red-Black Tree, B-Tree. From this survey, we can see that HEALPix based Binary Search on sorted arrays is the fastest and simple way to cross-match in memory, and the environment is easily restored from hardisk. If there are two unsorted catalogs on disk, a red-black balance tree with HEALPix indices would be a good choice. But when the catalog is too big to cache in memory, the memory-hardisk-swapping significantly slows down the efficiency. The key is keeping more points in the memory and do the Binary Search. Not only the speed to cross-match, how to efficiently exporting the rest data columns in catalogs is also considered. A cross-match web service built on these method is released.

1. Introduction

Cross-matching is widely used in Astronomy research. With this method, an astronomer collects different observations of a celestial object to learn its parameters in multi-wavelength and time-series sequences. Some applications, e.g., catalog offset calibration, object detection on image, also apply cross-matching.

Cross-matching is essentially based on astronomy catalogs. The size of catalogs could be small as several Kilobytes or even large than a Terabyte. Sky surveys, e.g., GAIA, SDSS, Pan-STARRS, are creating the huge catalogs. Many value added catalogs are also generated after more analyzing and calculation.

Normally, one row in a catalog belongs to an object or one observation. Cross-matching these catalogs is relative simple: to calculate objects' angular distance on the sphere. Besides the angular distance, other physical parameters could be combined to precisely identify one object, e.g., Red Shift, Proper Motion, Distance from the Earth etc. But not all catalogs contain these parameters, so angular distance is still the most important way to do cross-match.

This article will make a simple survey of current cross-matching methods. It is about two catalogs cross-matching, or we call it 2-Way cross-matching. Some other work also focuses on cross-matching multi-catalogs (N-Way cross-matching), but that is not the scope of this article.

2. Existing cross-matching method

Many cross-matching methods exist. Some of them directly apply angular distance formulas to calculate objects' angular distance one by one. For small catalogs, it is simple to implement and fast enough. Some other methods separate cross-matching into several stages to fast filter huge catalogs and detect the candidates.

Several formulas are widely used in cross-matching related works, e.g., the Great-circle formula¹, Haversine formula², and Vincenty formula³. The Vincenty formula is the most accurate and stable one, but it is too complicated. Great-circle is simple, but its deviation expands fast with distance. Haversine is a good alternative, to get a precise enough result and not quite complex to compute.

Input/Output bandwidth is a big problem to cross-match with big catalogs. Sorting data can avoid a whole table scan and significantly reduce I/O costs. But spherical coordinates is a two dimensional array; only indexing data in one dimension is not very efficient. So many pseudo two-dimensional spherical methods are invented to convert two dimension index to one, e.g., Zones Algorithm, HTM, HEALPix, Q3C et al. A bunch of software/library/services are based on these technologies to provide cross-matching function. We collect an incomplete list in the following.

- Zones Algorithm (Gray et al. 2007)
- Database extensions: PostgreSQL(Q3C, H3C, PostGIS, pgsphere), MySQL (mysql-sphere), SQLite (sqlite-sphere)
- Online cross-match service: CDS Xmatch (file based), SDSS SkyQuery (SQL online), GAVO Cross-Matcher Application
- Java package: STILTS, Topcat, Aladin
- Python library: smatch⁴, AstroPy⁵
- GPU cross-match code: <https://github.com/matthiaslee/Xmatch>

3. Code at hand is still important

Existing tools are all great. But sometimes, these tools cannot satisfy some requirement. For example: catalogs are too big, it takes too much time to import to database and build indices. Online services always limits the size of the uploaded dataset and return only a dozen columns of data. The internet bandwidth is always too narrow, it is impossible to upload Terabyte size of file to web service. Tools cannot be installed/run on Windows

¹Great-circle formula https://en.wikipedia.org/wiki/Great-circle_distance

²Haversine formula https://en.wikipedia.org/wiki/Haversine_formula

³Vincenty formula https://en.wikipedia.org/wiki/Vincenty%27s_formulae

⁴smatch <https://github.com/esheldon/smatch>

⁵AstroPy <https://www.sites.google.com/site/mrpaulhancock/blog/theage-oldproblemofcross-matchingastronomicalsources>

or other platform. When data gets too big, offline tools stack overflow and crash. Or cross-matching is part of the pipeline, we should have code at hand.

But not all the codes we need are open-source. So we have to reinvent the wheel, again. We have to choose the index key, the data structure, to determine how to index and how to search the data.

4. Performance test

In order to find the suitable method, we design a data flowchart in Fig.1 to test different indexing methods and data structures. Two catalogs are named as Sample Catalog and Reference Catalog. The Preprocessing step is to calculate the key: HTM ID, HEALPix ID, Zone ID, etc. At the Sort step, the Sample Catalog will be sorted by different data structures, e.g., Array, Red-Black Tree, 3-4 Tree, B tree, KD-Tree, etc. Different data structures will lead to different strategies to find candidates, then we precisely examine the distance to get final result.

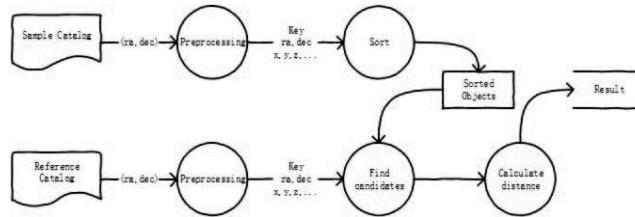


Figure 1. Flowchart to test index and data structure.

At first, we use HEALPix ID as the indexing key to test different data structures on different sizes of data. The result is shown in the left image of Fig.2. Array is simple and runs very well, and the environment is easily restored from hardisk. If there are two unsorted catalogs on disk, a red-black balance tree with HEALPix indices would be a good choice. But when the catalog is too big to cache in memory, the memory-hardisk-swapping slows down the efficiency a lot. For example, the B-Tree takes more than 1 day when catalogs more than 10^8 rows, and we cannot get the result. But B-Tree is not designed for memory based data query, it is the solution for hardisk based search. Many databases are using B-Tree, B+Tree, B*Tree to index data.

Based on Array, we apply different indexing keys, including HEALPix ID (*npix*), ZoneID of Zones Algorithm, and Dec-RA. "X-Y-Z" means the spherical distance calculation is based on Cartesian distance, and "binary" uses data in binary format but not CSV. From the middle image of Fig.2, apparently *npix* is the better indexing key. But which *nside* level of *npix* is proper? We make more tests in the right image of Fig.2. The image shows that *nside* = 12 is the best choice. But it is a rough result of the *nside* selection. For a different density of catalog, it is quite possible to fin another value. It might need more testing to find the best parameter for a specified catalog.

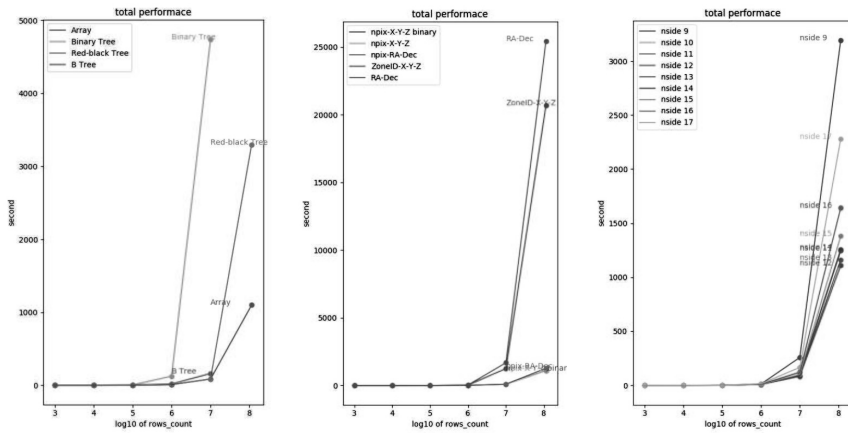


Figure 2. Left image shows the performance among different data structures. Middle image is different key's efficiency in Array. The influence of HEALPix's *nside* parameter is shown in right image.

5. Conclusion

Many existing tools and algorithms, e.g., Topcat, Astropy, Zones Algorithm, are very helpful. But sometime we need the code at hand to satisfy very special situation. A proper data indexing scheme and data structure needs to be determined. For the test of this paper, HEALpix ID and simple Array is a good choice. But current test is very limited, more tests should be done to include HTM, Q3C and other schemes. And more data structures are worth a try.

Cross-matching is much more intensive in Input/Output than CPU. The key to fast cross-match is trying to load everything in memory and do not swap memory and disk. If multi-node sources is available, e.g., map-reduce, Spark, CPU or GPU cluster, that will be very helpful. Array data structure is good and easy to extend to MPI/CUDA. Healpix ID also performs well, but the choice of *nside* level is still needed to find a way to determine. A simple cross-matching service based on current test is released at <http://xmatch.china-vo.org>.

Acknowledgments. This work is supported by National Natural Science Foundation of China (11503051, 11803055), the Joint Research Fund in Astronomy (U1531111, U1531115, U1531246, U1731125, U1731243) under cooperative agreement between the NSFC and Chinese Academy of Sciences, the 13th Five-year Informatization Plan of Chinese Academy of Sciences (No. XXH13503-03-107). We would like to thank the National R&D Infrastructure and Facility Development Program of China, "Earth System Science Data Sharing Platform" and "Fundamental Science Data Sharing Platform" (DKA2017-12-02-07). Data resources are supported by Chinese Astronomical Data Center and Chinese Virtual Observatory.

References

Gray, J., Nieto-Santisteban, M. A., & Szalay, A. S. 2007, eprint arXiv:cs/0701171. cs/0701171

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Extragalactic Stellar Photometry and the Blending Problem

Carlos Feinstein,^{1,2} Gustavo Baume,^{1,2} Jimena Rodríguez,^{1,2} and
 Marcela Vergne^{1,2}

¹*Facultad de Ciencias Astronómicas y Geofísicas, Observatorio Astronómico,
 UNLP, La Plata, Argentina; cfeinstein@fcaglp.unlp.edu.ar*

²*Instituto de Astrofísica de La Plata - CONICET*

Abstract. The images provided by the Advanced Camera for Surveys at the Hubble Space Telescope (ACS/HST) have the amazing spacial resolution of 0".05/pixel. Therefore, it is possible to resolve individual stars in nearby galaxies and, in particular, young blue stars in associations and open clusters of the recent starburst. These data are useful for studies of the extragalactic young population using color magnitude diagrams (CMD) of the stellar groups. However, even with the excellent indicated spatial resolution, the blending of several stars in crowded fields can change the shape of the CMDs. Some of the blendings could be handled in the cases where they produce particular features on the stellar PSF profile (e.g. abnormal sharpness, roundness, etc). But in some cases, the blend could be difficult to detect, this is the case, were a pair or several stars are in the same line of sight (e.g. observed in the same pixel). In this work, we investigated the importance of the blending effect in several crowded regions, using both numerical simulations and real ACS/HST data. In particular, we evaluated the influence of this effect over the CMDs, luminosity functions (LFs) and reddening estimations obtained from the observations.

1. Introduction and observations

As new high spatial resolution data have been obtained from nearby galaxies, some new problems have to be taken in account at the instance of analyzing the data. One of them is the blending, and more importantly, the blending affects on the observation parameters of the PSF (e.g two stars in same sight of view and without being detected). Some of the blending can be easily found analyzing sharpness, roundness, crowding parameters which are in the typical output of the photometry software. But, we want to focus on the case where the two stars are centered in the same pixel. So, is the undetected blending a real a problem?. By modeling it with data (density, stars colors, etc) from real observations of the ACS/HST we can check how probable is this happening in real data, how it would affect the observations and the results, for example, the CMD of the young clusters observed in nearby galaxies.

Data used in this work were obtained from the ACS/HST. They were collected and reduced by the ACS Nearby Galaxy Survey (ANGST - PI:Dalcanton). The selected data were download from the STSCI/ANGST site and they included photometric information for several fields covering the galaxies NGC 247, NGC 253, NGC 300 and NGC 2366. The ANGST sample was defined in (Dalcanton et al. 2009).

2. Probability of a blend

One way to estimate the number of blends is to make Monte Carlo simulations of a stellar population over a virtual CCD with spatial resolution and area equal to the real CCD on the ACS/HST, simulating real observations. Random stars are located using a uniform probability distribution, that has the maximum stellar density observed in the real ACS/HST Images. This maximum stellar density (see table 1) was obtained observing the real data from the galaxies to a certain magnitude. So, the blend of the sample can be easily computed in any simulations as the number of stars is chosen to reproduce the maximum density. In the real case, this number of blends would be valid at the place in the galaxy with the maximum stellar density and a maximum level for all regions with lower density (see for example, (Kiss & Bedding 2005)). But the same result can be more easily calculated because the uniform random distributions follow a Poisson spatial distribution over each pixel. Therefore its characteristic parameter is the stellar surface density (ρ) and the probability of having any star or one without a blend in a pixel is trivial to calculate. If N is the number of blends, the Possionian probability for a blend o more is $P(N > 1) = 1 - e^{-\rho}(1 + \rho)$.

So, stellar density is the key parameter, that would make that the stars in nearby galaxies are resolved individually or blended. **Table 1** shows the measured density in some galaxies of interest. The maximum density refers to the high value measured over the galaxy and its correspondance to the crowded central regions. On the other hand, the average density correspond to the disc typical density in each galaxy.

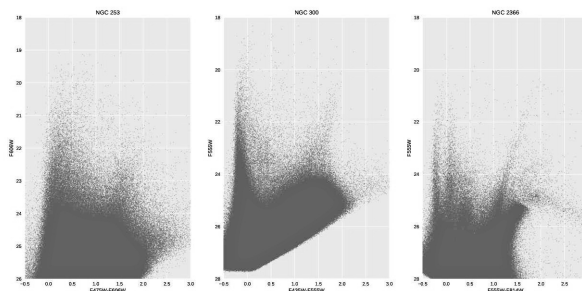


Figure 1. Color - Magnitude Diagrams. Note the young main sequence that indicates a young new population of stars recently formed

3. Self blends on the young blue population

In the core of a young cluster or association, where the density of blue stars is high, could the stars blend and make photometry programs measure several false stars that are brighter than the real ones? This could be a possibility, but in this case, in the top of the main sequence (MS) stars would show in comparison with ZAMS models to be too massive or would give a wrong estimation of distance. As we considered less massive and faint stars, a red fake component would be added and it could be confused with interstellar extinction. Again, we have not measured in any of these galaxies densities

that could make a real probability to have many blends (Table 1). But, stars that have a binary component are probable causing this kind of blends.

4. Color contamination

Fig.2 are the CMDs but showing the density of stars in the color-magnitude field normalized to the total number of stars. These plots shows that the majority of the observed stars are at mag. 27 or fainter. On the other hand, very few stars are on the MS track. As the stars are so faint, a blend with a MS massive star would not change its color or magnitude in an appreciable way. Three or four magnitudes of difference is at least one order of magnitude in energy flux.

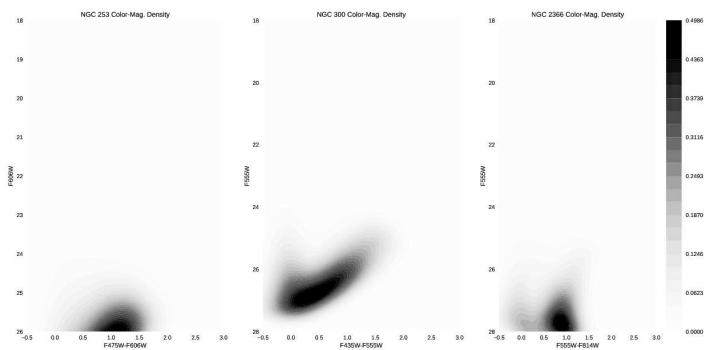


Figure 2. These are the Color - Magnitude Density Diagrams. Which is basically the same data as Fig. 1 but now normalize density of stars is show. Note the regions with high density are far away from the blue massive stars main sequence.

Table 1. Density of stars in the sample of galaxies considering in some cases a different limit magnitude. A limiting faint magnitude would let to find more stars, but these stars are too faint that any blends with then would not produce any observable effect. Density is measured in stars arcsec⁻²

Galaxy	Average Density	Max. Density	Filter	Limit Mag.	Distance Mpc	Max. Prob. of Blends
NGC 247	0.078	1.42	F606W	24	2.55	6.25 10 ⁻⁶
NGC 247	1.663	12.67	F606W	26	2.55	4.911 10 ⁻⁶
NGC 253	0.248	7.61	F606W	24	3.43	1.788 10 ⁻⁴
NGC 300	0.407	7.41	F555W	25	1.49	1.698 10 ⁻⁴
NGC 300	1.617	13.53	F555W	26	1.49	5.593 10 ⁻⁴
NGC 2366	0.064	0.76	F555W	24	3.93	1.812 10 ⁻⁶
NGC 2366	0.581	3.60	F555W	26	3.93	4.016 10 ⁻⁵

5. Blends and distance

It is easy to compute the inverse problem. This is knowing the star density, find at what distance the observations of individual stars would be blended. Considering the real case, where ACS/HST pixels are $0''.05$ wide, and the density of NGC 300 (for a magnitude limit of detection of $F555W < 24$). Fig. 3 shows close 9 Mpc, we are going to have a 20 % of blend contamination and this situation would be critical in the analysis of the data.

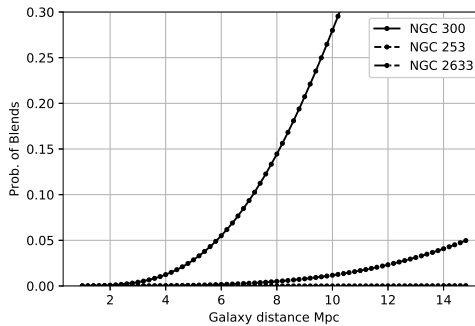


Figure 3. Prob. of a blend as the distance increases

6. Conclusions

- For the observed density on several galaxies, we found that the undetected blends are negligible in the nearby galaxies of our sample.
- For undetected blends to be important (for example, like 20% of contamination of the sample) the stellar density has to reach $0.0104 \text{ stars arcsec}^{-2} \text{ Mpc}^{-2}$ for most dense region in a galaxy. Considering that the pixel of the ACS/HST camera is $0''.05$ wide this would happen for a galaxy close to 9 Mpc in the case of NGC 300.
- The main source of undetected blends (which is not considered in this work) are the binaries systems. Binaries need time resolved spectroscopy to be detected and are an old problem not in extragalactic data but also on galactic observations.

References

- Dalcanton, J. J., Williams, B. F., Seth, A. C., Dolphin, A., Holtzman, J., Rosema, K., Skillman, E. D., Cole, A., Girardi, L., Gogarten, S. M., Karachentsev, I. D., Olsen, K., Weisz, D., Christensen, C., Freeman, K., Gilbert, K., Gallart, C., Harris, J., Hodge, P., de Jong, R. S., Karachentseva, V., Mateo, M., Stetson, P. B., Tavaréz, M., Zaritsky, D., Governato, F., & Quinn, T. 2009, *ApJS*, 183, 67. [0905.3737](#)
- Kiss, L. L., & Bedding, T. R. 2005, *MNRAS*, 358, 883. [astro-ph/0501498](#)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Astrophysical Code Migration into Exascale Era

D. Goz, S. Bertocco, L. Tornatore, and G. Taffoni

INAF - Osservatorio Astronomico di Trieste - via Tiepolo 11, 34131 Trieste
Italy; david.goz@inaf.it

Abstract. The ExaNeSt and EuroExa H2020 EU-funded projects aim to design and develop an exascale ready computing platform prototype based on low-energy-consumption ARM64 cores and FPGA accelerators. We participate in the application-driven design of the hardware solutions and prototype validation. To carry on this work we are using, among others, Hy-Nbody, a state-of-the-art direct N -body code. Core algorithms of Hy-Nbody have been improved in such a way to increasingly fit them to the exascale target platform. Waiting for the ExaNeSt prototype release, we are performing tests and code tuning operations on an ARM64 SoC facility: a SLURM managed HPC cluster based on 64-bit ARMv8 Cortex-A72/Cortex-A53 core design and powered by a Mali-T864 embedded GPU. In parallel, we are porting a kernel of Hy-Nbody on FPGA aiming to test and compare the performance-per-watt of our algorithms on different platforms. In this paper we describe how we re-engineered the application and we show first results on ARM SoC.

1. Introduction

The current market offers low-power micro-processor hardware solutions integrating enough transistors to include an on-chip floating-point unit capable of running typical HPC (High Performance Computing) applications. They are less expensive and more power-efficient than standard HPC devices. For this reason SoC (System on Chip) solutions are a possible approach to actually reduce the costs of HPC in terms of time and power consumption and this becomes extremely important when designing the new generation of HPC supercomputer, the exascale platforms. The ExaNeSt H2020 project (Katevenis et al. 2016) aims at the design and development of an exascale-class prototype computing system built upon power-efficient hardware able to execute real-world applications coming from a wide range of scientific and industrial domains, including also HPC for astrophysics (Ammendola et al. 2017). The ExaNeSt basic compute unit consists of low-energy-consumption ARM CPUs, FPGAs and low-latency interconnects (Katevenis et al. 2018).

Programmers will have to re-engineer their applications in order to fully exploit this new exascale platform based on heterogeneous hardware. We studied whether a direct N -body code for real scientific production may benefit from embedded GPUs given that the powerful high-end GPUs already have demonstrated to provide tremendous performance benefit for N -body code. To the best of our knowledge, this is the first work to implement such algorithm on embedded GPUs and to compare results with multi-core solutions on a SoC implementation.

2. Code implementation

Hy-Mbody is a direct N -body code that relies on the Hermite 6th order time integrator and that has been conceived to exploit hybrid hardware. The code is derived from HiGPUs (Capuzzo-Dolcetta *et al.* 2013; Capuzzo-Dolcetta & Spera 2013; Spera 2014), which has been widely used for simulations of star clusters with up to ~ 8 million bodies (Spera & Capuzzo-Dolcetta 2015; Spera *et al.* 2015), and of galaxy mergers (Bortolas *et al.* 2016). The kernels of Hy-Mbody have been developed with OpenCL in order to write efficient code for hybrid (CPU/GPU/FPGA) architecture. Kernels have been optimized using (i) *vectorization*, to increase the number of operations per cycle, and exploiting the (ii) *local memory* of the device, to reduce the latency of data transactions. The OpenCL host code is parallelized with hybrid MPI+OpenMP programming. A one-to-one correspondence between MPI processes and computational nodes is established and each MPI process manages all the OpenCL-compliant devices of the same type available per node. Inside of each shared-memory computational node, parallelization is achieved by means of OpenMP environment.

The Hermite 6th order integration schema requires double precision (DP) arithmetic in the evaluation of inter-particles distance and acceleration in order to minimize the round-off error. Full IEEE-compliant DP-arithmetic is efficient in available CPUs and GPGPUs, but it is still extremely resource-eager and performance-poor in other accelerators like embedded GPUs or FPGAs. The extended-precision (EX) numeric type is a valuable alternative in porting our application on devices not specifically designed for scientific calculations, such as embedded GPUs or FPGAs. We implemented in Hy-Mbody the EX-arithmetic as proposed by Thall (2006).

On SoC the memory is shared between CPU and GPU, so using local memory as a cache with associated barrier synchronization can waste both performance and power. For this reason, we implemented a specific embedded-GPU-optimized version of all kernels of Hy-Mbody.

3. Testbed description

We deployed a cluster based on heterogeneous hardware (CPU+GPU) to validate and test the Hy-Mbody code. Each computational node is a Rockchip Firefly-RK3399 single board computer. It is a six core 64-bit High-Performance Platform, based on SoC with the ARM big.LITTLE architecture. The main characteristics of this cluster, named INCAS¹, are listed in Table 1, while full details are in Bertocco *et al.* (2018).

4. Performance results

We just focused on the most computationally demanding kernel of the Hermite 6th order algorithm (with N bodies the kernel has $O(N^2)$ computational cost) and compared the performances on ARM CPUs. The left panel of Figure 1 shows the ratio of the best running time achieved by the CPUs as a function of the number of particles for both arithmetics. ARM Cortex-A72 with two cores is faster than Cortex-A53 with four cores by approximately a factor of two.

¹INTensive Clustered Arm-Soc

Table 1. The main characteristics of our cluster used to test the Hy-*N*body code.

Cluster name	INCAS
Nodes available	8
SoC	Rockchip RK3399 (28nm HKMG Process)
CPU	Six-Core ARM 64-bit processor (Dual-Core Cortex-A72 and Quad-Core Cortex-A53)
GPU	ARM Mali-T864 MP4 Quad-Core GPU
RAM	4GB Dual-Channel DDR3 (per node)
Network	1000Mbps Ethernet
Power	DC12V - 2A (per node)
Operating System	Ubuntu version 16.04 LTS
Compiler	gcc version 7.3.0
MPI	OpenMPI version 3.0.1
OpenCL	OpenCL 2.2
Job scheduler	SLURM version 17.11

High-end GPGPUs have already proved to speedup the solution of the direct *N*-body problem. In this work we aim to evaluate the performance of low-power embedded ARM GPU. We studied the best running time on ARM Cortex-A72x2 as the ratio over the best execution time taken by our ARM-optimized GPU implementation, as shown in the right panel of Figure 1. The ARM-optimized implementation is as fast as the dual-core implementation on the ARM Cortex-A72x2 using DP-arithmetic, as long as the GPU is kept fed with enough particles, while is almost three times faster using EX-precision.

5. Future development

ExaNeSt project is facing, among others, the challenge of the sustainable power consumption focusing on efficient hardware acceleration. For this reason, we are planning also to quantitatively measure the impact of our algorithms on energy consumption on SoC, shedding some light on their suitability for exascale applications. The findings from this research activity on ARM SoC are fundamental in order to also enhance our capabilities to exploit FPGAs for HPC, which in comparison to both CPUs and GPUs provide higher throughput-per-watt.

6. Conclusions

In light of our findings, embedded GPUs appear to be attractive from a performance perspective as soon as their double-precision compute capability increases. However, we demonstrated that the extended-precision approach can be a solution to supply enough power to execute scientific computation and benefit at maximum of the SoC devices.

SoC technology will play a fundamental role on future exascale heterogeneous platforms that will involve millions of specialized parallel compute units. Programmers will have to re-design their codes in order to fully exploit embedded accelerators, because of restricted hardware features compared to high-end GPGPUs.

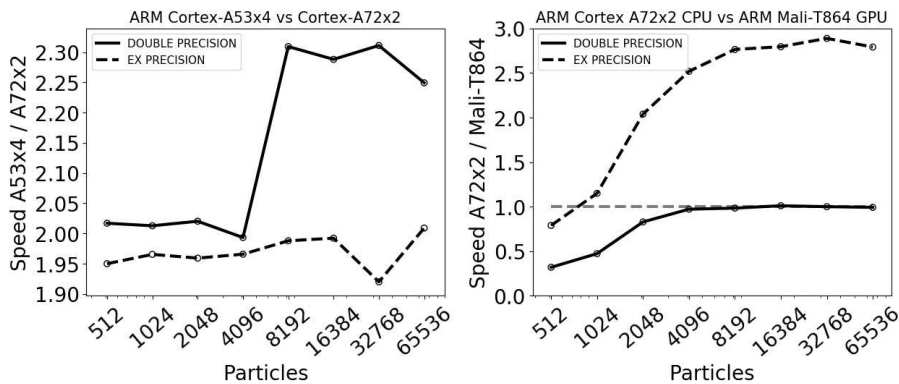


Figure 1. Left: Running time comparison between ARM Cortex-A53x4 and Cortex-A72x2 CPUs for both DP-arithmetic (solid line) and EX-arithmetic (dashed line) as a function of the number of particles. Right: comparison of the time to solution between ARM Cortex-A72x2 CPU and Mali-T864 GPU for both DP-arithmetic (solid line) and EX-arithmetic (dashed line) as a function of the number of particles.

Acknowledgments. This work was carried out within the ExaNeSt (FET-HPC) project (grant no. 671553) and the ASTERICS project (grant no. 653477), funded by the European Union's Horizon 2020 research and innovation program.

References

- Ammendola, R., Biagioni, A., Cretaro, P., Frezza, O., Cicero, F. L., Lonardo, A., Martinelli, M., Paolucci, P. S., Pastorelli, E., Simula, F., Vicini, P., Taffoni, G., Pascual, J. A., Navaridas, J., Lujan, M., Goodacre, J., Chrysos, N., & Katevenis, M. 2017, in 2017 Euromicro Conference on Digital System Design (DSD), 510
- Bertocco, S., Goz, D., Tornatore, L., & Taffoni, G. 2018, in INAF-OATs technical report, 222
- Bortolas, E., Gualandris, A., Dotti, M., Spera, M., & Mapelli, M. 2016, MNRAS, 461, 1023. 1606.06728
- Capuzzo-Dolcetta, R., & Spera, M. 2013, Computer Physics Communications, 184, 2528. 1304.1966
- Capuzzo-Dolcetta, R., Spera, M., & Punzo, D. 2013, J. Comput. Phys., 236, 580. 1207.2367
- Katevenis, M., Ammendola, R., Biagioni, A., Cretaro, P., Frezza, O., Lo Cicero, F., Lonardo, A., Martinelli, M., Paolucci, P., Pastorelli, E., Simula, F., Vicini, P., Taffoni, G., Pascual, J., Navaridas, J., Luján, M., Goodacre, J., Lietzow, B., Mouzakitis, A., Chrysos, N., Marazakis, M., Gorlani, P., Cozzini, S., Brandino, G., Koutsourakis, P., Ruth, J., Zhang, Y., & Kersten, M. 2018, Microprocessors and Microsystems, 61, 58
- Katevenis, M., Chrysos, N., Marazakis, M., Mavroidis, I., Chaix, F., Kallimanis, N., Navaridas, J., Goodacre, J., Vicini, P., Biagioni, A., Paolucci, P. S., Lonardo, A., Pastorelli, E., Cicero, F. L., Ammendola, R., Hopton, P., Coates, P., Taffoni, G., Cozzini, S., Kersten, M., Zhang, Y., Sahuquillo, J., Lechago, S., Pinto, C., Lietzow, B., Everett, D., & Perna, G. 2016, in 2016 Euromicro Conference on Digital System Design (DSD), 60
- Spera, M. 2014, ArXiv e-prints. 1411.5234
- Spera, M., & Capuzzo-Dolcetta, R. 2015, ArXiv e-prints. 1501.01040
- Spera, M., Mapelli, M., & Bressan, A. 2015, MNRAS, 451, 4086. 1505.05201
- Thall, A. 2006, in ACM SIGGRAPH 2006 Research Posters (New York, NY, USA: ACM), SIGGRAPH '06. URL <http://doi.acm.org/10.1145/1179622.1179682>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Pixel Mask Filtering of the CIAO Data Model

Helen He, Mark Cresitello-Dittmar, and Kenny Glotfelty
Harvard-Smithsonian Center for Astrophysics, Cambridge, MA 02138, USA

Abstract. The data model library (DM), a Chandra X-ray data analysis software (CIAO) package(Fruscione et al. 2006), facilitates a wide range of "on-the-fly" data manipulation capabilities such as copy, merge, bin and filtering. We have recently extended these capabilities to include pixel mask filtering. Pixel mask filtering is integrated into the DM region filtering syntax and logic, and thus complements the conventional geometric shape filtering. This integration allows for the standard operations on combining masks and shapes such as include, exclude, intersection, and union. The mask image is stored as a data block in the output file, and referenced by the region filter string in the data subspace. We will highlight the usage of pixel mask filtering in the CIAO software and discuss the unique features associated with the mask filtering.

1. What Is a Mask?

The data model library provides flexible filtering syntax for analyzing X-ray images or events. Pixel mask filtering extends the analytic shapes filtering capabilities. The syntax is denoted by 'mask(<file>)', where 'file' refers to a mask image file.

The mask file is a two dimensional (2D) image/table storing data in FITS or ASCII. The data type of a mask file can be any numerical type, which are recognized internally as byte value 0 or 1, where any non-zero values are treated as 1. Therefore, any images can be used as mask file and any two numeric columns of a binary table can be binned to serve as a mask image. As such, the masking is not limited only to spatial domain.

The mask is stored as part of the data subspace in the output, and always written as byte type images in 0 or 1 regardless of the input data type. The stored mask includes two parts: subspace region filter string, containing 'MASK(<block-name>)', and the corresponding data block, as illustrated in Table 1.

Table 1. Subspace Region Filter String and Mask Data Blocks

Region String		Data Blocks			
sky	TABLE MASK	Block 2: EVENTS	Table	15 cols x 985	rows
	MASK(MASK) MASK(MASK2)	Block 3: GTI3	Table	2 cols x 7	rows
	Field area = Region area =	Block 4: GTI1	Table	2 cols x 15	rows
		Block 5: MASK	Image	Byte(20x25)	
		Block 6: MASK2	Image	Byte(40x35)	

2. Mask Filtering

Data model filtering can be outlined in three formats but essentially falling into two categories, shapes and masks. Shape is parametric geometry, while mask is arbitrary geometry in pixels.

Because of its byte type, the mask filtering is to convolve mask image with events, so the events at some positions are screened (0) and others are passed (1). The resulting effect is to reduce the data noise/background or to remove some unwanted sources in the events. To apply a filter on a 2D 'sky' vector, the three filter formats are listed below.

- sky=shape(params): the shape, in general, can be any of the recognized shapes, 'circle', 'box', 'polygon', etc. A 'circle' filter is specified as 'sky=circle(x,y,r)', where 'x,y,r' are the circle's center position and radius. Only events in the 'circle'd region are passed and kept.
- sky=region(regfile): the tag, 'region()', instructs the DM library to read 'regfile', which stacks a variety of shapes with include or exclude operation. Therefore, this syntax builds shapes-combining filters.
- sky=mask(maskfile): the tag, 'mask()', instructs the DM library to read 'mask-file', which defines a pixel 'box' region in values 0 or 1 or in any numerical values.

Table 2 lists the mask filter syntax, along with region/shape filters for comparison. The 'f' in the 'mask(f)' and 'region(f)' represents mask and region file, respectively. The prescript 'exclude' of the filter syntax indicates the 'region' or 'shape' is excluded and the 'mask' region is reversed.

Mask	Region	Shape
sky=mask(f)	sky=region(f)	sky=circle(x,y,r)
exclude sky=mask(f)	exclude sky=region(f)	exclude sky=circle(x,y,r)
sky=bounds(mask(f))	sky=bounds(region(f))	sky=bounds(circle(x,y,r))
sky=mask(f1), det=mask(f2)	sky=region(f1), det=region(f2)	sky=circle, det=box
sky=mask(f1)&sky=box	-	-
sky=region(f2)	-	-

2.1. Masks Intersection in AND-Operator

Masks intersection can be illustrated simply by applying successive mask filters to a file. For example, using the CIAO tool, 'dmcoppy',

```
% dmcoppy 'evt[sky=mask(mask1)]' evt_m1
% dmcoppy 'evt_m1[sky=mask(mask2)]' evt.copy
```

Here, from the first command, 'evt_m1', carries a MASK extension, mask1.

The output, 'evt.copy', has a MASK block storing the data of 'mask1&mask2', and the resulting image is consolidated to the overlapping portion of the input masks.

The AND-Operation can be described through two **l_{xk}** matrix, say m1 and m2. The 'm1&m2' is to multiply each element of the two source matrices, put the result at the same position of the output matrix.

2.2. Masks Union in OR-Operator

The union operation can be illustrated by merging files containing MASK filters. The CIAO tool, 'dmmerge', merges event tables into a single output table. When events contain mask subspace/blocks, say 'evt1[mask1]', 'evt2[mask2]', 'evt3[mask3]', the masks are combined to one mask subspace/block in the output table.

```
% dmmerge  infile= 'evt1,evt2,evt3'  outfile=evt.merged
```

If all the masks overlap one another, the output, 'evt.merged', has one MASK data subspace, whose values are calculated in 'm1||m2||m3' (as described below) and whose mask bounds may be broader to enclose the input masks being union-ed. The full union of masks like this, however, may not always happen as discussed in following cases.

a) Partial Union: Should mask1 and mask3 be overlapped but mask2 is singled out, the merged output would have 2 mask subspace components as expressed below, where MASK is the union of mask1 and mask2, while MASK2 is the copy of mask3.

```
Component 1:  MASK(MASK)
Component 2:  MASK(MASK2)
```

b) No-union at all: Should the events have complex filters which cannot be combined, regardless masks condition, the merged output would have 3 mask subspace components, MASK, MASK2, MASK3, which are simply the copies of mask1, mask2 and mask3, respectively.

```
Component 1: MASK(MASK)
Component 2: MASK(MASK2)
Component 3: MASK(MASK3)
```

Similar to the AND-Operation in section 2.1, the OR-Operation (m1||m2) is to sum up each element of the two source matrices and normalized by the non-zero sum, put the result at the same position in output matrix.

3. Mask Binning

Events can be applied with both filtering and binning simultaneously, including mask filtering. In a masking-and-binning process, the binning still follows all the current CIAO rules. For example, the bin scale value must match that of an existing file (image only) or must be the multiples of the existing image's scale value. Uniquely, masks in mask filtering are also binned accordingly and stored for output. The algorithm of mask image re-binning is to take the logical AND-operation of the contributing pixel cells. In other words, if all contributing pixel values for the binned cell is 1, the binned pixel value is 1 otherwise it is 0.

3.1. Matching Coordinates

Generate a mask image file, 'mask.scale2', in bin scale factor 2. Create two sample events images in bin scales 1 and 2, 'img1' and 'img2':

```
% dmcoppy 'evt_1[bin sky=2]' mask.scale2
% dmcoppy 'evt[bin sky=1]' img1; dmcoppy 'evt[bin sky=2]' img2
```

Run dmcoppy on the events, img1 and img2, with the mask.scale2 filtering,

```
% dmcoppy 'img1[sky=mask(mask.scale2)]' error.img
% dmcoppy 'img2[sky=mask(mask.scale2)]' okay.img
```

The first run above exits with an error due to mis-matched coordinates of the mask (mask.scale2) from the events (img1). The second run succeeds since the coordinates of the mask and events (img2) match.

3.2. Bin Scale Multiples

Run dmcoppy events masking-and-binning with mask.scale2 file in various bin scales,

```
% dmcoppy 'evt[sky=mask(mask.scale2)][bin sky=2]' evt_es1.img
% dmcoppy 'evt[sky=mask(mask.scale2)][bin sky=4]' evt_es2.img
% dmcoppy 'evt[sky=mask(mask.scale2)][bin sky=6]' evt_es3.img
```

All the runs above are successful as the bin scales are the multiples of 2 (the mask's scale value). But, the following binning specs will cause errors as the bin scales are not the multiples of 2,

```
% dmcoppy 'evt[sky=mask(mask.scale2)][bin sky=3]' error.img
% dmcoppy 'evt[sky=mask(mask.scale2)][bin sky=5]' error.img
```

4. In Summary

Pixel mask filtering is a new and powerful addition to the CIAO DM library package, allowing users to filter not only on analytic shapes, but also on arbitrary shapes as well. The image convolution algorithm makes the mask filtering more effective than the conventional shape filtering. Mask filters can be applied to either spatial or non-spatial data, thus it is flexible and robust. Storing the mask image in the output file subspace allows us to keep track of the filtering history, so the data manipulation remain transparent. Lastly, the syntax of the masking application, similar to the well-known shape filtering, is familiar and easy.

Acknowledgments. The work has been supported by NASA under contract NAS 8-03060 to the Harvard-Smithsonian Center for Astrophysics for operation of the Chandra X-ray Center.

References

Fruscione, A., et al. 2006, in *Observatory Operations: Strategies, Processes, and Systems* (International Society for Optics and Photonics), vol. 6270, 62701V

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

The Algorithms Behind the HPF and NEID Pipeline

Kyle F. Kaplan,¹ Chad F. Bender,¹ Ryan C. Terrien,² Joe Ninan,³ Arpita Roy,⁴
 and Suvrath Mahadevan³

¹*University Of Arizona, Tucson, AZ, USA; kfkaplan@email.arizona.edu*

²*Carleton College, Northfield, MN, USA*

³*Pennsylvania State University, University Park, PA, USA*

⁴*California Institute of Technology, Pasadena, CA, USA*

Abstract. HPF and NEID are new high-resolution stabilized echelle spectrometers at the forefront of using precision radial velocity techniques to search for terrestrial mass exoplanets. High RV precision requires us to carefully consider the algorithms we use in our data reduction and analysis pipeline in order to avoid information loss, data smoothing, or algorithmic noise. We present a brief overview of a few of the algorithms we have chosen to convert unprocessed 2D echellograms into optimally extracted 1D spectra. These algorithms include the use of 2D interpolation that uses polygon clipping to rectify the curvature of the spectra across the detectors and the ability to fully account for aliasing in under-sampled data on the detector using flat lamp spectra as an empirical measurement of the cross-dispersion profiles used to weight our optimal extractions.

1. Introduction

The Habitable-zone Planet Finder (HPF) and NN-Explore Investigations with Doppler spectroscopy (NEID) are new mechanically and thermally stabilized high-resolution echelle spectrometers designed to search for terrestrial mass exoplanets with precision measurements of host star radial velocities (RVs). HPF covers 0.81–1.28 μm at a resolution of $R \sim 55000$ and has a photon limited RV precision goal of $\sim 1.0 \text{ m s}^{-1}$ (Mahadevan et al. 2012, 2014). It is installed on the 10 m Hobby-Eberly Telescope at McDonald Observatory. NEID covers 0.35–1.11 μm at a resolution of $R \sim 100000$ and has a RV precision goal of $\sim 10 \text{ cm s}^{-1}$ (Schwab et al. 2016). It will soon be installed on the 3.5 m WIYN Telescope at Kitt Peak National Observatory. Both spectrometers reside in temperature controlled rooms beneath each telescope and are fed by three optical fibers: one for the science target, another offset onto the sky, and a third for the simultaneous wavelength calibration source, such as a laser frequency comb (LFC) or emission lamp, to ensure accurate wavelength calibrations.

HPF and NEID achieve high RV precision through a combination of mechanical stability, accurate wavelength calibration, large wavelength coverage, high spectral resolution, and careful treatment of the data processing. We must carefully select the algorithms used in our data reduction and analysis pipeline to avoid unwanted smoothing, information loss, algorithmic noise, or systematic shifts in our results. In this conference proceeding, we discuss several algorithms that we are using in our pipeline,

focusing on the extraction of the dispersed spectral beams on the detector into fully calibrated 1D spectra made ready for RV measurements.

2. Rectifying the Beams With Polygon Clipping

In cross-dispersed spectrographs, the dispersed beams typically follow a curved path across a detector's quantized, regularly gridded, square shaped pixels. The cross-dispersion axis of the beam is not in angular alignment with the detector pixels. The window used to extract the beam also follows a similarly curved path across the detector. Ignoring these problems can lead to degradation in the spectral resolution and can introduce quantization into the localization of the extraction window. This quantization can lead to unwanted variation in the amount of background light inside the window or lost light in the beam wings at the window edges. This is important to consider if there is background light or if adjacent beams are close enough to cause cross-talk contamination. One solution is to forward model the spectrum and find the best fit model to the beams on the detector (Bolton & Schlegel 2010). The most commonly used solution is to rectify, or straighten, the curved beams before proceeding with the 1D extraction.

Rectification simultaneously resamples the spectrum and transforms it into a new reference frame of wavelength vs. cross-dispersion position. We want to minimize several sources of uncertainty introduced in the rectification process. Rectification always requires some form of interpolation, since the underlying shape of the spectrum is discretely sampled. Some common types of interpolation, such as polynomial interpolation, introduce “interpolation error” which arises when the chosen interpolation method does not fully reconstruct shape of the spectrum. The PSF will always contain some power above the Nyquist frequency, so even “perfect” band-limited interpolation methods will lead to aliasing or “ringing” for under-sampled data.

We have chosen the 2D interpolation method of polygon clipping to rectify the beams. It is commonly used in applications such as computer graphics, but is not commonly used for processing astronomical data. Polygon clipping has several advantages over other interpolation methods: it is flux conserving, minimizes resampling noise, can easily handle transformations, does not introduce “ringing” like band-limited interpolation, and minimizes correlations in the error of overlapping pixels. We adopt a variation of the technique outlined in Smith et al. (2007). The detector pixels are treated as square shaped polygons. Polygons representing pixels in the rectified reference frame are transformed into the detector reference frame and mapped over the polygons representing the detector pixels. The Sutherland & Hodgman (1974) method of polygon clipping is used to calculate the areas of each detector pixel that overlaps each pixel in the rectified reference frame. There are many ways to weight 2D interpolation. Polygon clipping calculates the flux in each rectified pixel to be the sum of the fluxes of the overlapping detector pixels weighted by their areas of overlap. The areas of overlap are computed from the clipped polygons. Figure 1 shows an example of beams from a spectrum taken with HPF rectified by polygon clipping.

3. Fixing the Problem of Under-sampled Beam Edges

HPF and NEID are spectrometers designed for exceptional image quality. Illumination variations on the fiber ends due to guiding and pupil changes during an observation

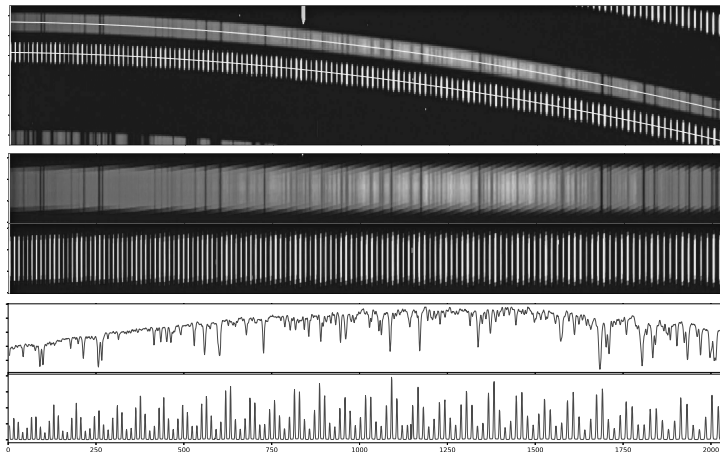


Figure 1. Top: Example of a single order from an HPF spectrum showing unrectified beams from a star in the science fiber and the LFC in the calibration fiber. The white line traces the center of the beams. Middle: Stellar and LFC beams rectified with polygon clipping. Bottom: Stellar and LFC beams collapsed into 1D using optimal extraction (Horne 1986).

introduce unwanted RV shifts. To avoid this problem, the spot size on the HPF and NEID detectors are less than a single pixel (Schwab et al. 2016), but a trade off of choosing to make the spot size so small is that the beam edges are under-sampled. The Nyquist theorem states that signal reconstruction requires sampling a signal at a rate that is at least twice as high as the highest frequency in the signal. The curvature of the beams across the regularly gridded detector pixels combine with the under-sampled beam edges to introduce aliasing as seen in Figures 1 and 2.

We use the optimal extraction algorithm (Horne 1986) to collapse our rectified 2D spectra into 1D spectra (see bottom of Figure 1). Optimal extraction is a summation method weighted not only by the uncertainty in the data but also by the cross-dispersion (vertical) profile. Aliasing in an under-sampled spectrum introduces a pattern with sharp features (bottom panel of Figure 2) that are not easily fit by a continuous smooth function or low pass filter such as a running mean. To fully characterize the cross-dispersion profile used to weight the optimal extraction, we have adopted the solution of using spectra of our flat lamp to directly measure the cross-dispersion profile (middle panel of Figure 2). The stability of the HPF and NEID spectrometers make this possible. To remove small scale variations on the detector, the flat lamp spectra are horizontally median smoothed. They are then rectified and each column is normalized and then used as the cross-dispersion profile to weight the optimal extractions which collapse our spectra into 1D.

4. What is Next?

We are committed to delivering the the best RV science results. While the algorithms presented here represent the current state of the HPF and NEID pipeline, our pipeline development process is ongoing and these algorithms will be continue to be evaluated,

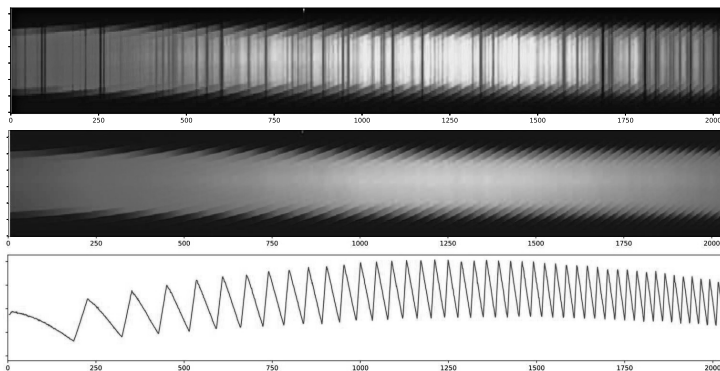


Figure 2. Top: Rectified beam for HPF order showing a stellar spectrum from the science fiber. The aliasing at the top and bottom of the beam where the beam edges are under-sampled is evident. Middle: Rectified beam showing flat lamp spectrum for the same order on the HPF detector as the stellar spectrum above. The same aliasing is evident. We use the flat lamp as the cross-dispersion profile for optimal extraction (Horne 1986). Bottom: Cross-section of a row near the top edge of the flat lamp beam showing the aliasing.

compared to alternative methods, and improved upon when possible. Periodic data releases for HPF and NEID are planned, and each release will rerun all data through the latest version of the pipeline.

Acknowledgments. I acknowledge the support from ADASS 2018 in the form of a travel grant, which helped enable me to attend this conference.

References

- Bolton, A. S., & Schlegel, D. J. 2010, *PASP*, 122, 248. 0911.2689
- Horne, K. 1986, *PASP*, 98, 609
- Mahadevan, S., Ramsey, L., Bender, C., Terrien, R., Wright, J. T., Halverson, S., Hearty, F., Nelson, M., Burton, A., Redman, S., Osterman, S., Diddams, S., Kasting, J., Endl, M., & Deshpande, R. 2012, in *Ground-based and Airborne Instrumentation for Astronomy IV*, vol. 8446 of *Proc. SPIE*, 84461S. 1209.1686
- Mahadevan, S., Ramsey, L. W., Terrien, R., Halverson, S., Roy, A., Hearty, F., Levi, E., Stefansson, G. K., Robertson, P., Bender, C., Schwab, C., & Nelson, M. 2014, in *Ground-based and Airborne Instrumentation for Astronomy V*, vol. 9147 of *Proc. SPIE*, 91471G
- Schwab, C., Rakich, A., Gong, Q., Mahadevan, S., Halverson, S. P., Roy, A., Terrien, R. C., Robertson, P. M., Hearty, F. R., Levi, E. I., Monson, A. J., Wright, J. T., McElwain, M. W., Bender, C. F., Blake, C. H., Stürmer, J., Gurevich, Y. V., Chakraborty, A., & Ramsey, L. W. 2016, in *Ground-based and Airborne Instrumentation for Astronomy VI*, vol. 9908 of *Proc. SPIE*, 99087H
- Smith, J. D. T., Armus, L., Dale, D. A., Roussel, H., Sheth, K., Buckalew, B. A., Jarrett, T. H., Helou, G., & Kennicutt, R. C., Jr. 2007, *PASP*, 119, 1133. 0708.3745
- Sutherland, I. E., & Hodgman, G. W. 1974, *Commun. ACM*, 17, 32. URL <http://doi.acm.org/10.1145/360767.360802>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Alpha-X: An Alpha Shape-based Hierarchical Clustering Algorithm

Ramsey L. Karim,¹ Lee G. Mundy,¹ and Isabelle Joncour^{1,2}

¹*University of Maryland, College Park, Maryland, USA; rkarim@astro.umd.edu*

²*Univ. Grenoble Alpes, CNRS, IPAG, F-38000 Grenoble, France*

Abstract. Alpha-X is an ongoing exploration into the utility of alpha shapes in describing hierarchical clustering. Based on the Delaunay triangulation of a set of points, alpha shapes describe concrete boundaries of regions around point clusters that are associated at distances less than some characteristic length scale α . The concept of alpha shapes is a discrete approach and can thus be applied to sets of positions of stars to evaluate stellar clustering. We are developing a representation of point-cloud substructures as α values in tree representations which capture the hierarchical lineage of structure at different values of α . With this approach, alpha shapes could be used in an new hierarchical cluster detection and characterization method that naturally defines boundaries associated with identified clusters.

1. Introduction

Alpha shapes, introduced in Edelsbrunner & Mücke (1994), offer a concrete, geometric definition of shape that can be used in a standardized approach to point-set analyses. The shapes generalize convex hulls in such a way as to describe non-convex organizations of points. The following section will introduce these concepts, but an in-depth formalism and more complete discussion of alpha shapes can be found in work by Edelsbrunner & Mücke (1994) and Edelsbrunner (2011).

1.1. Alpha Shapes

Algorithms to derive alpha shapes begin with the Delaunay triangulation of the point set and progress as a filtering of that graph. The Delaunay triangulation of an n -dimensional point set can be viewed as a simplicial n -complex, or collection of n -simplices. An n -simplex is the convex hull of $n+1$ points; a 1-simplex is a line segment, a 2-simplex a triangle, a 3-simplex a tetrahedron, and so on. A simplicial n -complex is generally a set of simplices whose dimension is no higher than n , but henceforth, any sets of simplices mentioned in this work are *homogeneous* simplicial n -complexes, which contain only simplices of dimension n (and, trivially, their faces).

A length scale can be associated with each simplex via its circumradius, the radius of the $(n-1)$ -sphere that circumscribes the n -simplex. The Delaunay triangulation can thus be filtered by a length scale parameter α to create the simplicial subcomplex known as the alpha complex, as illustrated by the progression in Figure 1. The alpha shape, outlined by solid blue lines in Figure 1, is the bounding face of the union of simplices

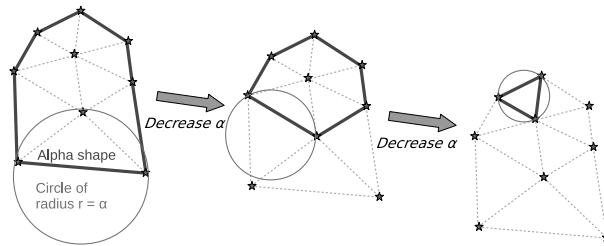


Figure 1. A simplicial 2-complex is filtered by progressively smaller α length scales, generating a series of alpha complexes whose boundaries are alpha shapes.

in this alpha complex. This shape is large at large α and shrinks as α is reduced and simplices are filtered out.

1.2. Alpha Shapes in the Astronomical Literature

The Delaunay triangulation has most notably appeared in the literature in the Delaunay Tessellation Field Estimator technique (Schaap & van de Weygaert 2000), which boasts a scale-invariant and parameter-free approach to local density estimation based on a discrete set of points. The Delaunay's dual graph, the Voronoi tessellation, is used more frequently, particularly for adaptive grids in computational applications.

Alpha shapes have been mentioned in work by van de Weygaert et al. (2010), which focused on topological analysis of cosmological simulation results and differs from ours in that we seek more specific shape quantifiers rather than large-scale notions. Outside astronomy, approaches to discrete point clustering for shape characterization using alpha shapes have been successfully implemented in fields such as molecular biology, pattern recognition, and digital shape sampling (Edelsbrunner 2011; Varshney et al. 1994).

2. Alpha-X Algorithm

The Alpha-X algorithm presented in this work is a divisive hierarchical clustering algorithm generalized to N dimensions, designed for sets of discrete positions of points in \mathbb{R}^N . The Delaunay triangulation of the point set yields a homogeneous simplicial n -complex. The volume and circumradius of any simplex in this set is calculated using the Cayley-Menger determinant (Hajja et al. 2017).

The algorithm represents a simplicial complex using a graph whose nodes are n -simplices and whose edges between nodes are $(n - 1)$ -simplices, shared faces of the n -simplices. This graph can be traversed to determine whether the set of simplices contains two or more disconnected subcomplexes. As α ranges from the maximum to the minimum circumradii associated with the triangulation, simplices are filtered out of the complex and the connectivity graph loses nodes. When the graph traversal indicates that the complex has fractured into disconnected subcomplexes, this process runs recursively on each subcomplex.

Each connected alpha complex represents a distinct cluster, such that a fractured cluster is considered to be the parent of the resulting subclusters. The algorithm starts with the original Delaunay complex, branching out when subclusters disconnect and

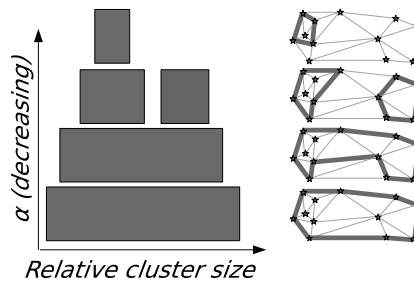


Figure 2. A modified dendrogram describes the hierarchical tree. Branch width indicates relative cluster size at a given vertical α value. Color marks cluster identity.

ending when α has filtered out all simplices and thus extinguished all independent clusters. The result is a hierarchical view of all points associated by n -simplices across the full dynamic range of length scales present in the set. This result is most naturally stored and traversed as a tree. We designed a modified dendrogram representation of this tree, illustrated in Figure 2 and exemplified in Figure 3. A collection of alpha complexes representing independent clusters can be retrieved for any α within the range of this tree, and each of these complexes is bounded by a well defined alpha shape.

Additional hyperparametric constraints may be imposed on the growing or completed tree in order to tune the result to the science goal and filter out uninteresting or noisy artifacts. A minimum size for tracked clusters extinguishes clusters before there is only one n -simplex left, since small clusters tend to demonstrate nothing more than the shot noise associated with sampling statistics. The parameter α can be reduced in uniform geometric steps, $\Delta\alpha \equiv \alpha_{i+1}/\alpha_i < 1$, whose size is another tunable parameter. Both of these parameters, given appropriate values, can reduce computation time and filter out noise artifacts without compromising the reliability of the result.

Notions of identity and persistence can also be contained in hyperparameters and can influence the interpretation of a hierarchy. A fractional subcluster size threshold, when exceeded, may indicate that the parent cluster should not be extinguished when it fractures into subclusters but should instead pass its identity to one of its children. Persistence is defined here as the range of length scales, or particularly, the number of $\Delta\alpha$ steps, between a cluster's α_i emergence from its parent and its $\alpha_f < \alpha_i$ extinction. The discussion of persistence by Edelsbrunner et al. (2000) highlights its utility for noise filtration, since cluster artifacts of shot noise are unlikely to persist across a wide range of α .

3. Scientific Applications

The Alpha-X clustering algorithm offers scale-invariant and parameter-free hierarchical clustering capability. The alpha shape approach works in \mathbb{R}^N but grows quickly in time and memory complexity. Its clustering ability is competitive with methods such as DBSCAN (Joncour et al. 2018), and it has the advantages of *a*) naturally defining a concrete boundary for each individual cluster at every step in the hierarchy and *b*) basing itself in a different field of mathematics than most other clustering algorithms,

offering utility as an uncorrelated comparison to another method. Figure 3 demonstrates this algorithm's performance on measured 2D sky coordinates of stars.

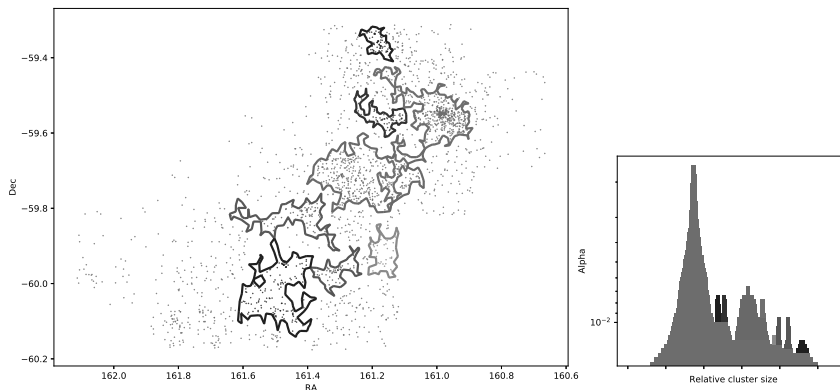


Figure 3. Alpha-X performance on the Carina Nebula star forming region. Alpha shapes are plotted around the included stars and color-matched to the dendrograms. *Left:* Alpha shapes plotted at the largest α level associated with each subcluster, and at the largest α level at which the main cluster (green) no longer has children. *Right:* The dendrogram associated with the hierarchical tree.

4. Conclusion

Alpha shapes show promise for use in a scale-invariant, parameter-free n -dimensional hierarchical clustering algorithm that offers naturally and concretely defined shape and boundary for each identifiable cluster. Our project will continue to develop and validate this methodology, comparing it to methods used within our field to verify that it offers novel and significant analysis products before moving to specific scientific applications.

References

- Edelsbrunner, H. 2011, in *Tessellations in the Sciences*, edited by R. van de Weijgaert, G. Vegter, J. Ritzerveld, & V. Icke (Springer Verlag), in press
- Edelsbrunner, H., Letscher, D., & Zomorodian, A. 2000, in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 454
- Edelsbrunner, H., & Mücke, E. P. 1994, *ACM Trans. Graph.*, 13, 43
- Hajja, M., Hammoudeh, I., & Hayajneh, M. 2017, *Beiträge zur Algebra und Geometrie / Contributions to Algebra and Geometry*, 58, 699
- Joncour, I., Duchêne, G., Moraux, E., & Motte, F. 2018, *A&A*, 620, A27
- Schaap, W. E., & van de Weygaert, R. 2000, *A&A*, 363, L29. [astro-ph/0011007](#)
- van de Weygaert, R., Platen, E., Vegter, G., Eldering, B., & Kruithof, N. 2010, in *Proceedings of the 2010 International Symposium on Voronoi Diagrams in Science and Engineering* (Washington, DC, USA: IEEE Computer Society), ISVD '10, 224
- Varshney, A., Brooks, F. P., & Wright, W. V. 1994, *IEEE Computer Graphics and Applications*, 14, 19

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Acceleration of the Sparse Modeling Imaging Tool for ALMA Radio Interferometric Data

George Kosugi,¹ Takeshi Nakazato,¹ and Shiro Ikeda²

¹*National Astronomical Observatory of Japan, Osawa 2-21-1, Mitaka, Tokyo 181-8588, Japan; george.kosugi@nao.ac.jp*

²*The Institute of Statistical Mathematics, Midori 10-3, Tachikawa, Tokyo 190-8562, Japan*

Abstract. Sparse modeling is widely used in image processing, signal processing, and machine learning recently. Thanks to the research and progress in statistical mathematics along with the evolution of computational power, the technique is applicable to the radio imaging for the data obtained with the ALMA (Atacama Large Millimeter-submillimeter Array). We've developed a new imaging tool based on the sparse modeling approach and it was experimentally implemented on the Common Astronomy Software Application (CASA) which is an official reduction software for the ALMA data. However, if the image size is large, e.g., 1K x 1K pixels, the data processing time gets longer, say several to ten hours, even with the latest mid-range server computers. Here we present a possible measure to greatly reduce the processing time.

1. Introduction

Radio interferometric imaging using the sparse modeling approach was originally developed for VLBI (Very Long Baseline Interferometry) observation data (Honma et al. 2014). Several simulated data were used to evaluate the method for years (Kuramochi et al. 2018). To automatically determine the most realistic solution from the infinite number of possible solutions, the cross-validation (CV) technique was introduced (Akiyama et al. 2017). We've developed a new interferometric imaging tool Python module for Radio Interferometry Imaging with Sparse Modeling (PRIISM, Nakazato et al. 2019) implemented on CASA, a standard data reduction application for ALMA data.

However, the new imaging technique with sparse modeling is computationally intense even for the latest CPUs. Furthermore, the CV process requires an order of magnitude more calculations. Reduction of its processing time is essential to utilize the technique with real ALMA data.

2. Process Overview

The most statistically realistic image can be derived by solving the following formula with the iterative process.

$$\mathbf{x} = \operatorname{argmin}[\|\mathbf{v} - F(\mathbf{x})\|^2 + \lambda_1 \|\mathbf{x}\| + \lambda_{TSV} TSV(\mathbf{x})] \quad (1)$$

subject to $\mathbf{x} \geq 0$, where \mathbf{x} is image, \mathbf{v} is visibility, $F()$ is Fourier transform, $TSV()$ is Total Square Variation function. Two regularization parameters λ_1 and λ_{TSV} control sparseness and smoothness, respectively. The equation can be solved iteratively, and the calculation is terminated when the image is converged. For each λ_1 and λ_{TSV} , we run the CV process to find the best probable (λ_1, λ_{TSV}) combination (Figure 1).

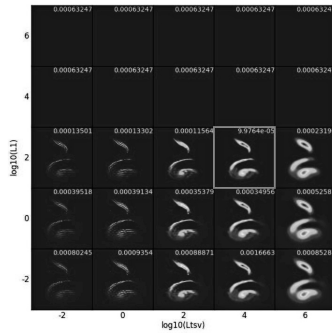


Figure 1. Sample chart resulting from the CV process. In this case, numerical values of 10^{-2} , 10^0 , 10^2 , 10^4 , and 10^6 were assigned to λ_1 and λ_{TSV} , and equation (1) was solved iteratively for every (λ_1, λ_{TSV}) combination. The most probable image can be obtained with $\lambda_1 = 10^2$ and $\lambda_{TSV} = 10^4$ (red square).

3. What's the Cross-Validation (CV) Process?

To choose the best probable regularization parameters λ_1 and λ_{TSV} , the CV process is introduced. In the CV process, visibility data is first divided into N (say 10) groups (N -fold CV), and then the process is run with $N-1$ groups of data (training set) to find a solution. We then apply the solution to the rest of the group (validation set) and calculate the deviation from the fit. The process is repeated with every combination of data groups; that means the process is repeated N times. Finally, we average all the deviation values derived by the process above. The smaller the averaged deviation is, the better the solution is presumed to be.

For every (λ_1, λ_{TSV}) combination, the whole CV process above is applied to determine the best λ_1 and λ_{TSV} , namely choose the combination of having the lowest averaged deviation. $\lambda_1 = 10^2$ and $\lambda_{TSV} = 10^4$ was selected in Figure 1 (in a red square frame). As one can imagine, this is really a CPU intensive process.

4. How to Accelerate the Process

Solving equation (1) is an iterative process. The resulting image is improved and converged gradually. Figure 2 shows how the image gets converged as the iteration increases. If the iteration cycle can be terminated earlier, the shorter the processing time goes. As is seen in Figure 2, the image is converged rapidly even in early cycles, and

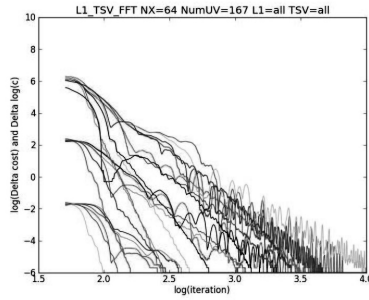


Figure 2. Image convergence curves for every $(\lambda_1, \lambda_{TSV})$ combination. The vertical axis represents the convergence: difference of the cost (in the square bracket of the right side of equation (1)) between adjacent iteration cycles. The lower the difference of the cost is, the more the image is converged. The horizontal axis shows the number of iteration cycles with a logarithmic scale.

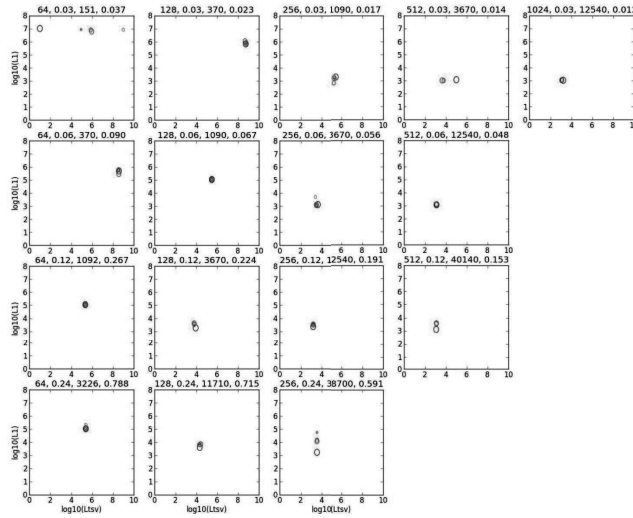


Figure 3. The most probable $(\lambda_1, \lambda_{TSV})$ combination derived from the CV process was plotted by changing the iteration cycle from 100 to 10000. From top to bottom, spatial resolution was changed from 0.03 to 0.24 arcsec/pixel. From right to left, size of the image was changed from 64x64 to 1024x1024. Ovals represent the CV result at a certain number of iteration cycles: largest Oval for 100, next largest oval for 300, middle sized oval for 1000, smaller oval for 3000, and the smallest oval for 10000 iterations. Numbers put on each boxes are image size (pixel), spatial resolution (arcsec/pixel), number of data on the uv-plane, and filling factor of the uv-plane, respectively.

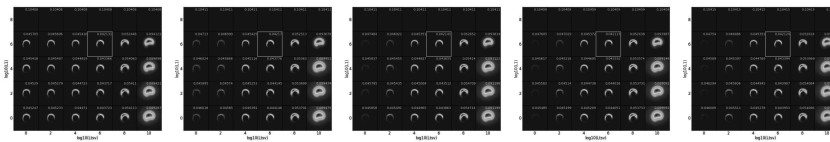


Figure 4. CV process with different iteration cycles: from left to right, 100, 300, 1000, 3000, and 10000 iterations, respectively. The vertical axis and the horizontal axis for each chart indicate λ_1 and λ_{TSV} , respectively.

in some cases it rebounds and oscillates in later cycles. But in general, the curve shows decreasing trend globally.

Figure 3 shows how the derived most probable λ_1 and λ_{TSV} are moved as the number of iteration cycle increases. In most cases, only small changes can be seen even if the iteration cycle is increased, and therefore, 100 iteration cycles is barely acceptable. For more safety, 300 or 1000 iteration cycles is enough for rough estimation of λ_1 and λ_{TSV} in the CV process.

Resulting charts from the CV process with different iteration cycles are shown in Figure 4. Most probable $(\lambda_1, \lambda_{TSV})$ combinations selected by the CV process were independent of the number of iteration cycles and identical in this case. The processing time was 29, 85, 279, 820, and 2740 sec for 100, 300, 1000, 3000, and 10000 iteration cycles, respectively. The iteration process was stopped when the difference of the cost between two adjacent cycles became smaller than a certain threshold value. Since the threshold is arbitrarily set, the value tends to be smaller so as to continue the iteration process until the image is fully converged. It requires a long time. However, if we split the whole process into two, namely, a light weighted CV process (small iteration cycle) only to determine the regularization parameters $(\lambda_1, \lambda_{TSV})$ combination and the final imaging iteration process (until the image is fully converged) with the derived parameter set $(\lambda_1, \lambda_{TSV})$, the whole processing time is greatly reduced. That is one of the practical solutions to accelerate the interferometric imaging process using PRIISM.

References

- Akiyama, K., Ikeda, S., Pleau, M., Fish, V. L., Tazaki, F., Kuramochi, K., Broderick, A. E., Dexter, J., Mościbrodzka, M., Gowanlock, M., Honma, M., & Doeleman, S. S. 2017, *AJ*, 153, 159. 1702.00424
- Honma, M., Akiyama, K., Uemura, M., & Ikeda, S. 2014, *PASJ*, 66, 95
- Kuramochi, K., Akiyama, K., Ikeda, S., Tazaki, F., Fish, V. L., Pu, H.-Y., Asada, K., & Honma, M. 2018, *ApJ*, 858, 56. 1802.05783
- Nakazato, T., Ikeda, S., Akiyama, K., Kosugi, G., Yamaguchi, M., & Honma, M. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 143

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Tensor Clusters for Extracting and Summarizing Components in Spectral Cubes

Mauricio Solar, Humberto Farias, and Camilo Nunez

Universidad Tecnica Federico Santa Maria, Santiago, Chile

mauricio.solar@usm.cl, humberto.farias@usm.cl,

camilo.nunezf@sansano.usm.cl

Abstract. Modern Integral Field Spectrograph instruments like the Multi Unit Spectroscopic Explorer generate large-scale data. The data size problem has been studied in recent times but there is another equally relevant problem, the data cubes also exhibit unprecedented complexity and dimensionality. These complex sources can be modeled using tensor algebraic methods. In this representation the cube has two physical axes (Ra/Dec), and a 3rd axis (Wavelength, Time, Intensity). The representation is a tensor of order 3 that leads to a concise description astronomical data cubes. We applied over this representation a tensor-clusters approach to find the morphology of components expected in these complex data cubes. To achieve this goal the TensorFit, a library with strong GPU support to handle spectral cubes in a tensor mode, was used.

1. Introduction

Integral field spectrographs (IFS) is one of the most powerful observation techniques to obtain spatially resolved spectroscopy of extended objects. IFS combining imaging and spectroscopy provides a powerful technique to provide a sharper and deeper 3D understanding of galaxies. The final product of IFS is a data-cube, with two spatial dimension (Ra/Dec) and a one spectral dimension (wavelength/velocity). The high density of information in a datacube, especially in cubes created using the techniques of Fibers+lenslets and Image slicer allow spatially resolved kinematics and emission lines in distant galaxies. IFS instruments like the Multi Unit Spectroscopic Explorer (MUSE) generate large-scale data. This will be accentuated with new ground-based telescope instruments such as HARMONI and METIS (E-ELT); and space telescopes such as JWST with MIRI. Therefore, unsupervised methods of analysis that address the multidimensional complexity of the IFS cubes are necessary.

2. IFS 2D analysis methods

There are two dominant (Bacon & Monnet 2017) methods for analyzing IFS cubes, the 1st approximation corresponds to a spectral analysis as the classical way of working these cubes using software such as IRAF (Fitzpatrick 1993). This spectral approach analyzes each spectrum in a sequence individually. The 2nd method is based on a spatial approach where individual images (monochromatic) are generated from the cube and on each of them software such as SExtractor is used to identify individual sources. Both methods work the spectral axis or the spaxels independently, without considering the

possible spatial relationships in these axes. Another example is the Zurich Atmosphere Purge (ZAP)(Soto et al. 2016) which is a sky subtraction tool for MUSE cubes. ZAP uses Principal Component Analysis (PCA) to calculate the eigenspectra and eigenvalues to isolate the residual sky subtraction features and remove them from the observed datacube (Bacon et al. 2015). Clearly these approaches are powerful tools having delivered outstanding results. Some authors (Bacon & Monnet 2017; González-Gaitán et al. 2018) propose an approach to address the global multidimensionality of the relationships on these spectral cubes.

3. IFS multidimensional (spatial) methods of analysis

As indicated in section 2, there is a need for new approaches to model the globality of the relationships present in the IFS cubes, i.e. find a cluster of spaxels with different emission lines generated by the same source. In a 2D approach if the spaxels are processed independently these relationships can be lost if the emission lines have a higher difference in Signal-to-noise ratio. In addition, the analysis of the datacube should consider this spatial covariance between adjacent spaxels. Therefore, we need multidimensional modeling when integrating multidimensional data. A spectral cube can be considered as a 3D representation or a multidimensional array from the computer science perspective. The analysis requires to consider the nature characteristics of its elements or the scientific objectives for which it was generated. An IFS cube must be analyzed without altering its geometric structure, i.e. it cannot be rotated. Must conserve its spatial coordinates and especially its spectral dimension. From the above we can infer that it requires methods of analysis that capture and preserve the multidimensional relationships of the information contained in these cubes. The aim of this proposal is to model the IFS cube as a 3-way tensor, using a multilinear algebra approach. To consider the operation of spectral cubes as a multidimensional problem, not as a simple spaxels stack. This approach allows to have the following advantages regarding techniques such as PCA used in ZAP:

- The data of an IFS cube are positive (Ivezić et al. 2014). This allows to apply non-negativity constraints.
- A tensor decomposition under certain conditions is unique. In particular as they are tensors of order 3 (Zhou et al. 2013). In other words, the clusters found are not exposed to the rotational problem of matrix factorization that PCA suffers.

The proposal is to move from linear methods of factorization (PCA) to multilinear methods of decomposition; that is, to use as a framework the theory of tensors to find the latent multi-linear manifold present in these astronomical data. Instead of the matrix approach, the present work focuses on the decomposition of the tensor. Specifically in CANDECOMP/PARAFAC (CP) that in broad terms decomposes a tensor as a sum of rank-one tensors and is in many fields a standard methods for unsupervised multidimensional data analysis.

3.1. CANDECOMP/PARAFAC (CP) Decomposition

It is also known as Tensor Rank Decomposition, since in essence it can be defined as the search for a compact representation of a tensor in the form of the sum of rank-1

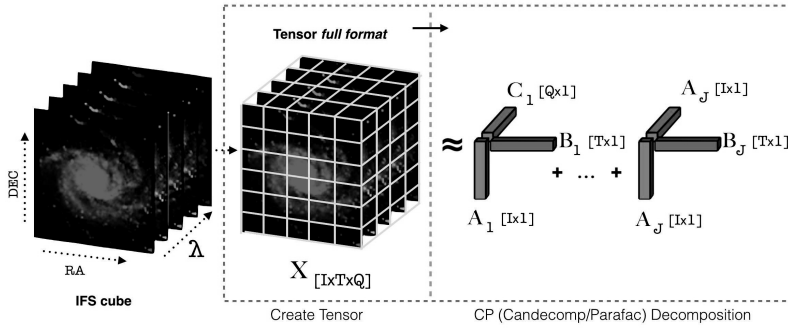


Figure 1. Pipeline of tensor decomposition in TensorFit. IFS cube represented as third-order tensor $X \in \mathbb{R}^{6 \times 5 \times 4}$

tensors, as we can see in figure 1. CP unlike PCA does not establish an orthogonality constraints to ensure its uniqueness, but it comes from the fact that CP decompose a tensor into a sum of rank-1 tensors.

Let $X \in \mathbb{R}^{I \times T \times Q}$ be a third-order tensor. The goal is to compute a CP decomposition with J components that best approximates X , i.e., to find equation 1, where each element of the resulting tensor is obtained by following equation 2.

$$\min_{\hat{X}} \|X - \hat{X}\| \text{ with } X = \sum_{j=1}^J A_j \circ B_j \circ C_j = [A, B, C]. \quad (1)$$

$$X_{i,t,q} = \sum_{r=1}^J A_{i,r} \circ B_{t,r} \circ C_{q,r}. \quad (2)$$

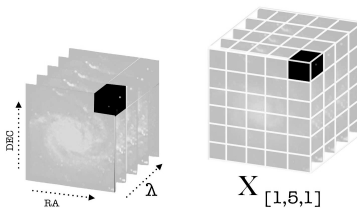


Figure 2. Voxel represented as sub-tensor, $X_{1,5,1}$

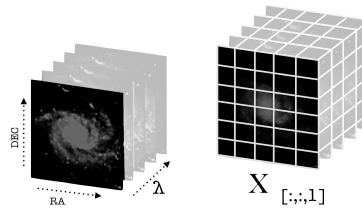


Figure 3. Spaxel represented as Mode-3 (tube) fibers, $X_{1,5,:}$

4. Experimental setting and Result

The experiment uses an observation of L1448, a star-forming region in Perseus, widely used in the validation of astronomical software, such as Astropy (Robitaille et al. 2013).

This data cube has a dimensions 105x105x54 (Ra, Dec, Velocity). CP was applied with different parameters using TensorFit (Farias et al. 2018) a library generated in a previous work of the authors. The integrated HI intensity map was calculated for the IFS cube and on this map are graphed different rank-1 tensor corresponding to spaxel clusters.

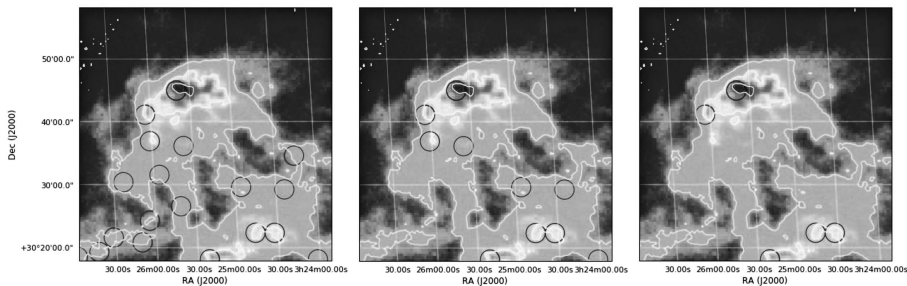


Figure 4. The integrated HI intensity map of the IFS cube of L1448

In Figure 4, on the first intensity map 19 rank-1 tensor were plotted. It is possible to verify that many of these do not intensify intensity clusters correctly. But by decreasing the number of these rank-1 tensors the identification improves significantly.

5. Conclusion

Although the results of this work are preliminary they require a detailed analysis. We can suggest that it is possible and feasible to model the IFS cubes as tensors, and on these apply CP to find spaxel clusters. These clusters are obtained from an analysis that addresses the multidimensional nature of the data present in the IFS cubes.

Acknowledgments. This research was possible due to the CONICYT-Chile funds, specifically through the project FONDEF IT15I10041 and Conicyt PIA/Basal FB0821.

References

- Bacon, R., Brinchmann, J., Richard, J., Contini, T., Drake, A., Franx, M., Tacchella, S., Vernet, J., Wisotzki, L., Blaizot, J., et al. 2015, *Astronomy & Astrophysics*, 575, A75
- Bacon, R., & Monnet, G. 2017, *Optical 3D-spectroscopy for Astronomy* (John Wiley & Sons)
- Farias, H., Nuñez, C., & Solar, M. 2018, *Astronomy and Computing*
- Fitzpatrick, M. J. 1993, in *Astronomical Data Analysis Software and Systems II*, vol. 52, 472
- González-Gaitán, S., de Souza, R., Krone-Martins, A., Cameron, E., Coelho, P., Galbany, L., Ishida, E., et al. 2018, arXiv preprint arXiv:1802.06280
- Ivezić, Ž., Connolly, A., Vanderplas, J., & Gray, A. 2014, *Statistics, Data Mining and Machine Learning in Astronomy* (Princeton University Press)
- Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., et al. 2013, *Astronomy & Astrophysics*, 558, A33
- Soto, K. T., Lilly, S. J., Bacon, R., Richard, J., & Conseil, S. 2016, *Monthly Notices of the Royal Astronomical Society*, 458, 3210
- Zhou, H., Li, L., & Zhu, H. 2013, *Journal of the American Statistical Association*, 108, 540

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Robust Registration of Astronomy Catalogs

Fan Tian, Tamás Budavári, and Amitabh Basu

Department of Applied Mathematics & Statistics, The Johns Hopkins University, Baltimore, Maryland 21218, USA

Abstract. Due to a small number of reference sources, the astrometric calibration of images with a small field of view is often inferior to the internal accuracy of sources detected in the images. One important experiment with such challenges is the Hubble Space Telescope (HST). A possible solution is to cross-calibrate overlapping fields instead of just relying on standard stars. Following Budavári & Lubow (2012), we use infinitesimal 3D rotations for fine-tuning the calibration but re-formalize the objective to be robust to a large number of false candidates in the initial set of associations. Using Bayesian statistics, we accommodate bad data by explicitly modeling the quality which yields a formalism essentially identical to M -estimation in robust statistics. Our preliminary results on simulated catalogs show great potentials for improving the HST calibration.

1. Motivation

Unlike large survey projects such as Sloan Digital Sky Survey (SDSS) designed to provide a catalog, HST images are often taken under specific programs surveying patched sky regions over different time domains. It is then essential to create a high-precision astrometry for Hubble sources to utilize the HST data. Although cross-calibrating large images to the World Coordinate System (WCS) standards can be done efficiently (Lang et al. 2010), lacking enough reference stars, small images such as those of HST are more challenging to work with. A novel approach taken was the practice in creating the Hubble Source Catalog (HSC; Whitmore et al. 2016) using the algorithms described in Budavári & Lubow (2012). By rotating images in 3D, they were able to cross-calibrate sources within Hubble to obtain an improved relative astrometry. With the number of standard stars increased in the aligned images, it also increases the chance to further crossmatch the astrometrically corrected images to larger referencing catalogs.

To align the many overlapping HST images, Budavári & Lubow (2012) introduce a 3D infinitesimal rotation vector, which measures the axis and the angle of the rotation for an image. The shifts are determined by minimizing the separations between paired sources. This approach essentially arrives at the optimization of a quadratic cost function. The algorithm works effectively when the initial image offset is small, but the issue raises for large residuals that can overpower small values in estimation. The current solution to this problem in HSC is to pre-determine approximately matched pairs using the *pre-offsets* method and a Bayesian likelihood comparison approach (Whitmore et al. 2016; Budavári & Lubow 2012). In this paper, we aim to provide a new approach that is free from the step of pre-defining nearly matched pairs. Borrowing tools from robust statistics, we can determine the registration for very large residuals

presented in the initial set of associations. In the following sections, we describe the new method in a simple scenario of cross-calibrating two images. Our implementation and its applications are performed on realistic simulations to the HST images.

2. Bayesian Formalism

Before carrying out alignment, we first determine a set of initial associations for a given *search radius* R from the positional data D of two images. By thresholding on separation, we can exclude obvious bad matchings to optimize estimation accuracy and to improve computation efficiency.

Next, we fix one image and correct the other to a reference direction relative to the first image. A reasonable choice of the reference calibrators are the midpoint directions of the matched pairs. Let the total number of pairs within R to be N_{pairs} . For $q \in \{1, \dots, N_{\text{pairs}}\}$, we represent the q -th source direction as \mathbf{r}_q with the corresponding calibrator direction as \mathbf{c}_q . Suppose we have the source-calibrator pairs with small residuals are the “good” members potentially forming the true associations, and those pairs with large residuals are the “bad” members, we represent the “good” and “bad” member likelihood functions as $\ell_q^G(\omega)$ and $\ell_q^B(\omega)$ respectively for ω to be the 3D infinitesimal rotation vector. Using γ to denote the probability of a pair being from a true association, with a prior knowledge on γ , ω is estimated from the joint likelihood function of

$$L_\gamma(\omega) = \prod_q [\gamma \ell_q^G(\omega) + (1-\gamma) \ell_q^B(\omega)]. \quad (1)$$

In practice, the member likelihood function $\ell(\omega)$ is often assumed to be Gaussian. Here we choose the 3D analog to the Gaussian distribution - the Fisher (1953) distribution - to describe the directional uncertainty of a unit vector \mathbf{r} on the sphere. The good member likelihood function is then $\ell_q^G(\omega) = F(\mathbf{c}_q; \mathbf{r}'_q(\omega), \kappa)$, with the concentration parameter $\kappa = 1/\sigma^2$ for small σ uncertainty. The transformation of \mathbf{r}_q sources by rotation vector ω is given by $\mathbf{r}'_q = \mathbf{r}_q + \omega \times \mathbf{r}_q$. The bad pairs follow an isotropic distribution with $\ell_q^B(\omega) = \frac{1}{4\pi}$, which is the case when $\kappa \rightarrow 0$ in Fisher.

Since γ is generally unknown in practice, we follow the discussion in Budavári & Loredó (2015), also see Budavári & Szalay (2008), and choose to guesstimate γ . For N_1 and N_2 being the number of sources in two catalogs respectively, we estimate γ with

$$\gamma_* = \frac{\min(N_1, N_2)}{N_{\text{pairs}}}. \quad (2)$$

Later we also find that our method is robust to the choice of γ , which making the use of Eq (2) practical. But one can always refine the estimation after finding the true associations in the corrected catalogs.

With the estimated probability γ_* and the member likelihood functions, we optimize the joint likelihood function in Eq (1) for ω and obtain the following objective function:

$$\tilde{\omega} = \arg \max_{\omega} \prod_q \left[\frac{\gamma_*}{2\pi \sigma^2} \exp \left\{ -\frac{|\mathbf{c}_q - (\mathbf{r}_q + \omega \times \mathbf{r}_q)|^2}{2\sigma^2} \right\} + \frac{1-\gamma_*}{4\pi} \right]. \quad (3)$$

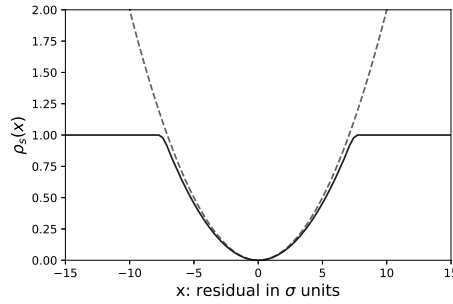


Figure 1. The robust ρ -function (solid blue line) limits the influence of outliers in comparison to a quadratic objective (dashed red line).

3. Connection to Robust Statistics

When all pairs are good, i.e., $\gamma_* = 1$, the above likelihood maximization yields the least-squares problem for ω as introduced by Budavári & Lubow (2012). As the fraction of good pairs decreases, the effective likelihood function gains heavier tails making the optimization more difficult. For any given γ_* we can take the negative logarithm of the likelihood and arrive at

$$\tilde{\omega} = \arg \min_{\omega} \sum_q \rho \left(\frac{|\Delta_q - \omega \times r_q|}{\sigma} \right) \quad \text{with} \quad \rho(t) = -\ln \left(\frac{2\gamma_*}{\sigma^2} e^{-t^2} + 1 - \gamma_* \right) \quad (4)$$

and $\Delta_q = c_q - r_q$. As illustrated in Figure (1), this ρ -function is quadratic for small residuals, but constant for large values - limiting the contribution of bad pairs to the objective. We note that ρ is a function of t^2 only and this problem formally is much like M -estimation in robust statistics (Maronna et al. 2006). We can solve this type of problems by an iterative procedure alternating between (1) solving for ω using $A\omega = b$ with

$$A = \sum_q \frac{w_q}{\sigma^2} (I - r_q \otimes r_q) \quad \text{and} \quad b = \sum_q \frac{w_q}{\sigma^2} (r_q \times c_q)$$

assuming constant w_q weights, and (2) re-evaluating those weights based on ω as

$$w_q = W \left(\frac{|\Delta_q - \omega \times r_q|}{\sigma} \right) \quad (5)$$

with $W(t) = \rho'(t)/t$. We find this procedure converges quickly in practice.

4. Results and Discussion

We implemented both the least squares estimation and the new robust method in a set of simulated images with different offsets. Since the ground truth is known, we directly measure the image offset using the median separation of true pairs. The correction accuracy is reported by comparing the initial image offset and the offset after correction.

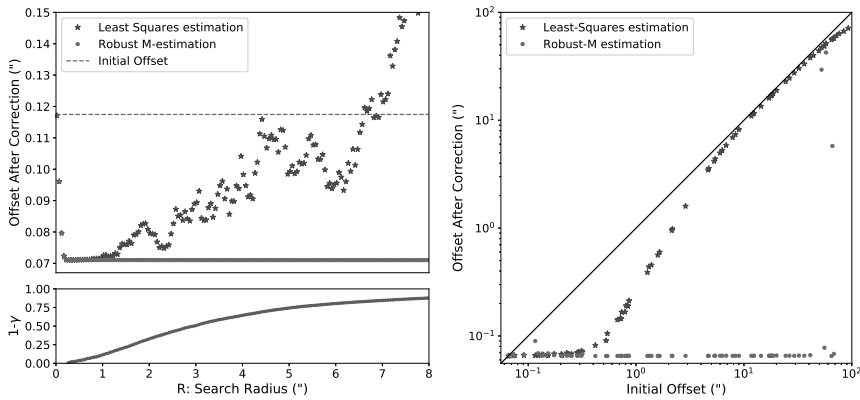


Figure 2. Comparison of our method with least squares estimation. Left top: test one pair of catalogs with increasing R ; dashed grey line indicates the small initial offset. Left bottom: probability of bad matchings. Right: test on catalog pairs with increasing initial offset

We compare both methods in two ways. As shown in Figure (2), the left top panel shows a comparison for two images with a small initial offset of approximately $0.1''$. Increasing the *search radius*, both methods recover the correct rotation when $R < 1''$. For $R > 1''$, the least squares method starts to break down. The robust M -estimator, on the other hand, can find the accurate rotation vector under large R . The left bottom panel measures γ as the fraction of the number of true pairs to the number of pairs within R . This reinforces the fact that the least squares estimation is less robust to extreme residuals. The right panel of Figure (2) compares the methods' accuracy for images with large offsets. To draw a fair comparison, we applied a *search radius* to be just a few σ above the initial offset. With small initial offsets ($< \sim 0.3''$), both the robust estimator and the least squares estimator correct the astrometry to approximately σ . As the initial offset increases to above $0.3''$, the least squares algorithm fails to find a correction. The robust M -estimation is accurate for offsets up to $60''$. Beyond $60''$, neither method can satisfactorily recover the rotation. To find correction under very large offsets, pre-determining likely associations is still preferred.

With the success on simulations, we are currently testing the new algorithm on HST data. Our future plan also involves cross-registering HST sources to the new data release of the *Gaia* telescope.

References

- Budavári, T., & Loredó, T. J. 2015, *Annual Review of Statistics and Its Application*, 2, 113
- Budavári, T., & Lubow, S. H. 2012, *The Astrophysical Journal*, 761, 188
- Budavári, T., & Szalay, A. S. 2008, *The Astrophysical Journal*, 679, 301
- Fisher, R. S. 1953, *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 217, 295
- Lang, D., Hogg, D. W., Mierle, K., Blanton, M., & Roweis, S. 2010, *AJ*, 139, 1782
- Maronna, R. A., Martin, R. D., & J., Y. V. 2006, *Robust statistics theory and methods* (J. Wiley)
- Whitmore, B. C., et al. 2016, *AJ*, 151, 134

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Development of Auto-multithresh: an Automated Masking Algorithm for Deconvolution in CASA

Takahiro Tsutsumi,¹ Amanda Kepley,² Ilsang Yoon,² and Urvashi Rau¹

¹*National Radio Astronomy Observatory, Socorro, NM, USA;*
ttsutsum@nrao.edu

²*National Radio Astronomy Observatory, Charlottesville, VA, USA*

Abstract. A general purpose automated masking algorithm for deconvolution was developed in order to support automated data processing in ever-increasing data volumes of the current and future radio interferometers as described by Kepley et al. in this conference (O12-1). In this paper, we describe some technical details of the implementation of the automated masking algorithm named, “auto-multithresh”, which was integrated into the refactored imaging task (TCLEAN) in CASA. We also discuss our approach that we took for the development, which loosely follows the iterative model, so that the implementation is refined progressively for its functionality and performance based on testing and updated requirements throughout prototyping in Python to the final production in C++.

1. Auto-multithresh Algorithm

A basic concept of this algorithm is to mimic interactive masking done by experienced astronomers during CLEAN. A user can control the parameter setting through the quantities such as *rms* noise, sidelobe level, and synthesized beam size. Figure 1 shows some of the key features of the process. More detailed description can be found in the CASA Docs¹ as well as Kepley et al. (in preparation).

The following are the key features of the algorithm.

- Iterative (run at the beginning of minor cycle).
- Threshold based mask created using a current residual image. An appropriate threshold value is chosen from user-specified parameters, which relate to the quantities such as *rms* noise and sidelobe level.
- “Prune” : mechanism to remove unrealistic (or noise like) mask regions, i.e., regions smaller than a user-specifiable parameter, which is in a fraction of the synthesized beam.
- “Grow”: grow the threshold based mask to include low surface brightness regions using a binary dilation algorithm.

¹e.g. for CASA 5.4.0 documentation: <https://casa.nrao.edu/casadocs/casa-5.4.0/synthesis-imaging/masks-for-deconvolution>

- Handle negative (absorption) and positive (emission) features. It tracks the two features separately to avoid interaction, then they are added together at the end of the process.
- For cube imaging, allow to skip channels for no mask or no mask change from the previous iteration.

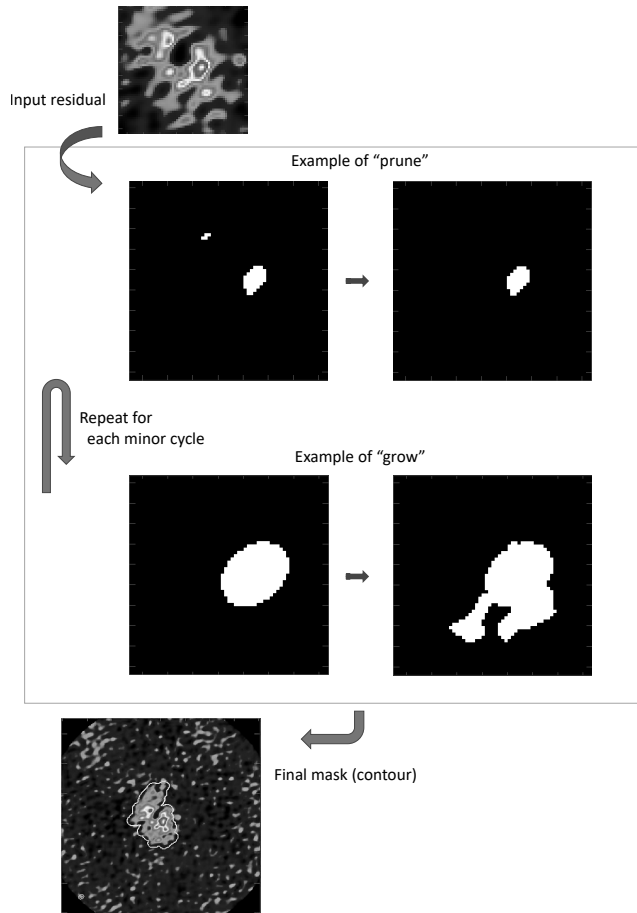


Figure 1. Auto-multithresh process (showing only some of the key features)

2. Implementation Details

The prototype algorithm development was done in Python². The modular design of the refactored imaging code (C++ and Python) in CASA (McMullin et al. 2007) allows

²<https://github.com/aakepley/autobox>

us to do flexible implementations. Figure 2 shows the code structure of the refactored imager. A wrapper Python class, PySynthesisImager, is built on the top of the collection of the synthesis imaging Python tools. The `tclean` CASA task provides a task interface to all the imaging tools provided through PySynthesisImager (see Figure 2). Since each of the tools has one-to-one mapping of C++ classes and methods, prototyping by Python scripts can be easily accomplished using PySynthesisImager.

The final implementation of the algorithm was done in C++. The auto-multithresh algorithm is implemented in a general mask handler class inside the refactored imaging code. It is launched from deconvolver to be in sync with its iteration control.

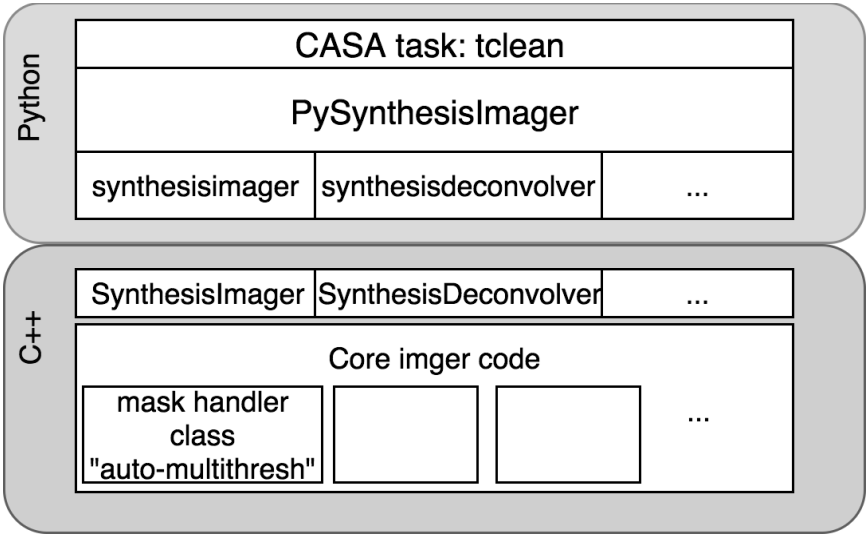


Figure 2. Implementation of auto-multithresh in CASA.

3. Development Process

The main driver of this development came from the ALMA pipeline. There was a research aspect to explore and refine an algorithm that work for the real ALMA data while meeting various time constraints including the CASA release schedules. It was necessary to adopt a development process slightly different from the standard CASA development which generally completes within a single CASA development cycle. To do this we had a team of a dedicated developer for implementation and a scientist who led in design and verification as well as other testers for additional scientific verification. The process of this development followed a loosely iterative development model.

- 1. Define requirements
- 2. The initial prototype development
- 3. Initial implementation

4. Verification/Validation Testing (performance, scientific correctness)
5. Amend or add to the requirements, if necessary
6. Implementation of additional features, mitigation to performance issues
7. Repeat steps 4 -6

The adopted process generally worked well to deliver of the necessary functionality on time with flexibility of adding new features or making corrections in next iterations. However, one of the disadvantages was that the significant dedicated time by the key members for both code development and verification/validation testing was required. As for future projects of this nature, the observatory is making an effort to plan to separate resources for production from R&D efforts.

4. Current Status

The algorithm has been available since the CASA 5.0 release in `TCLEAN` CASA task and various improvements were made ever since. It has been used in the production ALMA pipeline for Cycle 5 and beyond. While the original motivation was to be able use in ALMA imaging, it has been shown that the algorithm works on the data from other telescopes such as JVLA and ATCA.

5. Future Development

For CASA 5.5 release, we plan to complete bulk of the development, including a new noise estimate to improve masking of absorption and extended emission. As a future research, we plan to explore to improve code efficiency by moving a part of the algorithm deeper inside the deconvolution algorithms.

References

McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in ADASS XVI, edited by R. A. Shaw, F. Hill, & D. J. Bell (ASP), vol. 376 of ASP Conf. Ser., 127

Session XIII

Miscellaneous

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Receiving Credit for Research Software

Alice Allen^{1,2}

¹*Astrophysics Source Code Library, USA; aallen@ascl.net*

²*University of Maryland, College Park, MD, USA*

Abstract. Though computational methods are widely used in many disciplines, those who author these methods have not always received credit for their work. This presentation covered recent developments in astronomy, including new journals, policy revisions by existing journals, new and expanded community resources, changes to infrastructure, and availability of new workflows that make recognizing the contributions of software authors easier. The talk provided steps code authors can take to increase correct citation of their software and steps researchers can take to improve their articles by including citations for the computational methods that enabled their research.

1. Introduction

Science overall depends on software (Morin et al. 2012); informal surveys have demonstrated how thoroughly researchers depend on software in astronomy (Momcheva & Tollerud 2015) and in other disciplines (Hettrick et al. 2014). Indeed, someone stated at the 2018 European Week of Astronomy and Space Science meeting that “software is the most used instrument in astronomy.”¹

I started editing the Astrophysics Source Code Library (ASCL, ascl.net) in 2010; since taking on this role, I’ve seen shifts in astronomy and indeed in research in general that make it not only possible but increasingly easier for research software authors to receive credit for their work. This presentation covered some of these changes and offered advice for receiving and giving recognition to those who write the software upon which our discipline depends.

2. New journals and revised policies in existing journals

Over the previous eight years, astronomy and science research in general has seen an increase in the number of journals focused on software, with at least one journal coming online per year that focuses on or actively seeks to publish research software articles, or even the software itself.

These journals fill different niches and have different attributes, from those that are astronomy-specific (A&C, ComAC, RNASS) to those that are not discipline-specific (JORS, SoftX, JOSS), open access (JORS, ComAC, JOS, RNAAS) or closed or hybrid

¹I do not know who it was that said this; if you do, please let me know!

Table 1. New journals and year of first publication

Year	Title	Bibstem
2012	<i>Journal of Open Research Software</i>	JORS
2013	<i>Astronomy and Computing</i>	A&C
2014	<i>Computational Astrophysics and Cosmology</i>	ComAC
2015	<i>Software X</i>	SoftX
2016	<i>Journal of Open Source Software</i>	JOSS
2017	<i>Research Notes of the AAS</i>	RNAAS

access (A&C, SoftX), and those that accept more traditional articles about software (JORS, A&C, ComAC, RNASS) or essentially accept only the software with a very brief narrative (SoftX, JOSS), and even as to whether they are referred or not, with all but RNAAS being refereed. What is refereed and access to the software also varies; for SoftX and JOSS, the software itself is peer-reviewed and must be publicly available.

The list of journals above is not exhaustive; other journals, such as *Nature Astronomy* and *Computing and Software for Big Science*, have also recently entered the marketplace and welcome computational methods articles.

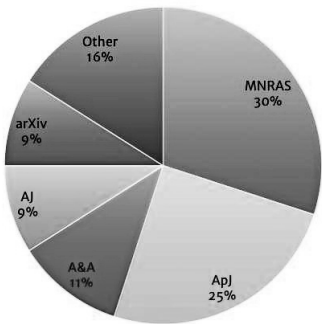


Figure 1. Citations to ASCL entries by journal as of 12/17/18

Some journals, such as *The Astrophysical Journal*, actively discouraged formal citations for software; these journals also typically had a policy that they would not accept articles that were primarily about software. These policies were not rigorously enforced, as is evident by citations to ASCL entries that appeared in these journals.² During the past eight years, these restrictive policies were dropped, bringing all major astronomy journals to parity in accepting both software articles and formal citations to software, as can be seen by the list of nearly 90 journals with citations to ASCL entries on the ASCL dashboard.³

These changes occurred for many reasons, including pressure from researchers, increasing concern about reproducibility and transparency (Peng 2011; Morin et al.

²The first ASCL entry citation was in *The Astrophysical Journal* in 2012 (Tollerud et al. 2012)

³<http://ascl.net/dashboard>

2012), interest in software citation and formal guidelines from NSF,⁴ NASA (NASA 2014), and other funders and organizations such as Force11 in its Software Citation Principles (Smith et al. 2016).

3. Community resources, infrastructure changes, and new workflows

New services have become available in the past decade, including collaborative coding sites such as Bitbucket and GitHub and archival resources such as Figshare and Zenodo. Existing services such as the ASCL have been given new life and are growing; in its Next Generation project, arXiv is improving its support for linking data and code to research.⁵ Software citation is captured, tracked, and counted by indexers such as ADS, Web of Science, and Google Scholar. Broader efforts that cross disciplines and influence not just astronomy but many fields include the Force11 organization, which published the aforementioned Software Citation Principles that an increasing number of journals are adopting. The sharing of ideas influences not just those involved in the efforts, but has a greater reach with their aspirational and practical goals and guidelines.

4. Actions for software authors

The steps below can improve having your code used and cited, thus getting you recognition for your software work.

Release your code. Make your software publicly available. Releasing software improves the transparency of research, and more transparent science is better science; it also improves citation and encourages collaboration.

Specify how you want your software cited. Be specific in how you want your software cited and make this information easy to find by putting it in your README, on the code's home page, in a citation file using the citation file format (CFF) standard (Druskat et al. 2018) or a codemeta.json file (Jones et al. 2017). To comply with the Force11 Software Citation Principles and ensure citation tracking in ADS, request citation for the software itself via ASCL ID or a DOI from an archival service, such as Zenodo or Figshare. Though citing an article using or describing the code results in an ADS-trackable citation and journals do accept this citation method, citing only an article does not meet Force11 Software Citation Guidelines. Do not use standard URLs for citing software! Use code site URLs in footnotes, in text, but not as a formal citation; use these only in addition, not instead of, a formal citation.

Assign a license. Assigning a license to your software lets other people know what they can do with your code, whether they can use it, adapt it, or incorporate it into additional projects. Help for choosing a license is readily available online.

Register your code. Register your code with the ASCL; this gives your program a unique identifier, creates entries for your software in ADS and other indexing services such as Web of Science, and provides a trackable citation method that is compliant with the Force11 Software Citation Principles. Your software is more discoverable not only from indexing, but also because ADS can link software with research papers that use

⁴<https://www.nsf.gov/pubs/2014/nsf14059/nsf14059.jsp>

⁵<https://confluence.cornell.edu/display/arxivpub/2018+arXiv+Roadmap>

it and vice versa through its *Associated articles* feature. The Submissions link on the ASCL's home page can be used to submit a code for inclusion in the ASCL.

Archive your code. Ensure your code remains available to complete the research record by archiving it in your university's library system, with the ASCL, or in services such as Zenodo, Figshare, Dryad, and the Open Science Framework.

5. Actions for researchers

Improve your research articles by including citations for the computational methods that enabled your research. To cite other people's codes well, look for the software's preferred citation information on the code download site, in the README or documentation, or on the ASCL. If you cannot find it, ask the author of the software and if that fails, check ADS to see how others cited the code, or submit the code to the ASCL to have an ASCL ID created for it. Include a "software" section in your paper in addition to (not instead of!) formally citing the software in the references. And finally, when you referee a paper, insist on proper citations for all of the codes used for the research.

6. Conclusion

Astronomy has seen changes in infrastructure, attitudes, expectations, and journal practices that make receiving credit for research software easier. Astronomers have numerous ways to increase recognition of software and those who author it by leveraging these recent changes.

References

- Druskat, S., et al. 2018, Citation file format (cff) - specifications. URL <https://doi.org/10.5281/zenodo.1405679>
- Hettrick, S., et al. 2014, UK Research Software Survey 2014. URL <https://doi.org/10.5281/zenodo.1183562>
- Jones, M. B., et al. 2017, Codemeta: an exchange schema for software metadata. URL <https://doi.org/10.5063/schema/codemeta-2.0>
- Momcheva, I., & Tollerud, E. 2015, arXiv e-prints. 1507.03989
- Morin, A., et al. 2012, *Science*, 336, 159
- NASA 2014, NASA Plan for Increasing Access to the Results of Scientific Research, Tech. rep. URL [http://www.nasa.gov/sites/default/files/atoms/files/206985_2015_nasa_plan_for_web.pdf](http://www.nasa.gov/sites/default/files/atoms/files/2069852015_nasa_plan_for_web.pdf)
- Peng, R. D. 2011, *Science*, 334, 1226
- Smith, A. M., et al. 2016, *PeerJ Computer Science*, 2:e86
- Tollerud, E. J., et al. 2012, *ApJ*, 752, 45. 1112.1067

Starting Up a Data Model for Exoplanetary Data

Marco Molinaro,¹ Eleonora Alei,^{2,5} Serena Benatti,² Andrea Bignamini,¹
François Bonnarel,³ Mario Damasso,⁴ Mireille Louys,³ Michele Maris,¹ and
Valerio Nascimbeni^{2,5}

¹*INAF – Astronomical Observatory of Trieste, Trieste, Italy;*

marco.molinaro@inaf.it

²*INAF – Astronomical Observatory of Padova, Padova, Italy*

³*CDS – Strasbourg astronomical Data Center, Strasbourg, France*

⁴*INAF – Astrophysical Observatory of Torino, Torino, Italy*

⁵*Department of Physics and Astronomy, University of Padova, Padova, Italy*

Abstract. The effort for searching, studying and characterizing extrasolar planets and planetary systems is a growing and improving field of astrophysical research. Alongside the growing knowledge on the field, the data resources are also growing, both from observations and numerical simulations. To tackle interoperability of these data, an effort is starting (under the EU H2020 ASTERICS¹ project) to delineate a data model to allow a common sharing of the datasets and collections of exoplanetary data. The data model will pick up model components from the IVOA specifications, either existing or under investigation, and attach new ones where needed. Here are presented the first results in drafting the exoplanetary systems dedicated data model. Relationships are reported with existing and proposed IVOA models; new key components not yet available in the interoperable scenario are shown. The results here reported cover a first set of requirements and considerations and take into account aspects like the observations of exoplanetary systems, the usage of existing exoplanets catalogues, the investigation of atmospheres of confirmed exoplanets and the simulation of exoplanet's atmospheres devoted to characterize exoplanets habitability.

1. Introduction

Starting from the experience aimed at deploying through the Virtual Observatory (VO) the Time Series for exoplanets (see, e.g., Molinaro et al. 2019), indication was given that a model to describe exoplanetary systems might be needed. This model would allow proper dataset discovery and description through metadata annotation. This need was later pointed out also by requirements from exoplanetary atmosphere numerical simulations, exoplanets catalogue investigation and visual client applications.

¹Astronomy ESFRI & Research Infrastructure Cluster: <https://www.asterics2020.eu/>

To tackle this modelling solution a first meeting was organized² to start gathering requirements and designing the first blocks (classes) of the model and to try to identify interests in the community to improve and adopt it. This contribution reports the outcome of that meeting: Sec. 2 reports the requirements brought in by the meeting participants, while Sect. 3 describes the steps taken in starting the modelling effort.

2. Requirements from the community

The initial requirements from the Global Architecture of Planetary Systems (GAPS) project³ (see, e.g., Benatti et al. 2016) mainly dealt with its time series of radial velocities of exoplanetary systems, both from optical spectroscopy and the subsequent efforts in joining it with its near-infrared counterpart. This clearly connects to the Time Series Data Model⁴. However, exoplanets characterization is done also by combining those radial velocity datasets with photometric ones. In both cases, disentangling stellar and planetary contribution in the observation result is required, pointing to the need to model stellar hosts. This can be tackled referencing the ongoing Source DM⁵ effort at IVOA. The continuation of the GAPS long term program will also see the study of exoplanets atmospheres, thus bringing in demands also on this aspect of the scenario to be modelled.

Further prerequisites were taken into consideration from the simulations of planetary atmospheres to characterize exoplanets habitability (ARTECS project, see Murante et al. 2018). ARTECS already made an attempt at using VO technologies to publish their results. The attempt led to identifying the main concepts and blocks in the exoplanetary systems description they use. The internal metadata model used in ARTECS shows clear connections to concepts available from Simulation Data Model (SimDM, Lemson et al. 2012), DataSet Metadata Model (Bonnarel et al. 2016) and Cube Data Model (Tody et al. 2015) with, possibly, the above mentioned Source DM. It also pointed to other components that are not yet available and were used for the model we will describe in Sec. 3.

Other requirements came from an ongoing effort in merging three main catalogues of exoplanets: Extrasolar Planets Encyclopaedia⁶, Exoplanet Orbit Database⁷, and NASA Exoplanets Archive⁸. The difficulties emerging when combining them range from the curation point of view (differing from catalogue to catalogue) but also in terms of accessibility, column annotation and bibliographic referencing for the various measurements reported. They clearly represent a use case where a model with common metadata and vocabularies would ease the effort in consuming different catalogues sharing the same domain.

²<https://www.asterics2020.eu/dokuwiki/doku.php?id=open:wp4:wp4exodm>

³http://www.oact.inaf.it/exoit/EXO-IT/Projects/Entries/2011/12/27_GAPS.html

⁴M. Louys talk at <https://wiki.ivoa.net/twiki/bin/view/IVOA/InterOpNov2018TDIG>

⁵J. Salgado talk at <https://wiki.ivoa.net/twiki/bin/view/IVOA/InterOpMay2017-DM>

⁶<http://exoplanet.eu/>

⁷<http://exoplanets.org/>

⁸<https://exoplanetarchive.ipac.caltech.edu/>

General discussion led also to identify constraints to modelling, e.g. to consider parameters that belong not to one celestial object, but to a couple of them, like the orbital elements or resonance periods. Finally, some other details touched the discovery scenario, where time constraints and stellar activity indexes went side by side with provenance information and planetary architecture descriptions (like “*system having a Jupiter and a Super Earth in an inner orbit*”).

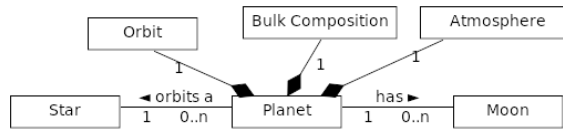


Figure 1. Starting model sketch, having the exoplanet as the central class.

3. Modelling efforts

Starting from the experiences described in Sec. 2, a basic model of planetary system data has been sketched like in Figure 1. The preliminary basic model above does not include the classes, derived from the atmosphere simulations, that detail the experiment description and datasets output. Those will be respectively referenced from the existing SimDM and DataSet Metadata - Cube - TimeSeries data models of the IVOA. Another point of contact with the IVOA works is represented by the Star class, which is considered to be a reference to the ongoing Source DM.

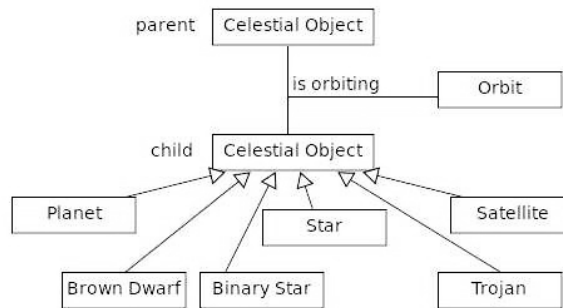


Figure 2. Refined initial model having the Orbit as a class associating a couple of Celestial Objects. Sub-classes are used to specialise objects.

The main point to move from a model like the one in Figure 1 to the more general solution in Figure 2 is the fact that an orbit is a concept related to a couple of Celestial Object(s), one of which is considered the child and the other the parent one. This solution, besides solving the flaw of considering the Orbit as a component to a Planet (or a single object), helps in directly considering satellites and other objects in the general scenario.

Sub-typing the Celestial Object class defines the various actors in the (exo-)planetary system scenario allowing flexibility for data to come in the near future. Also,

the connection to the Source DM is preserved. An open point is the one about a binary stellar system as the planetary system's host (ternary or multiple-star hosts are not currently the case). There are cases where planets orbit one of the stars in the binary system, and there are cases where planets orbit the common binary center of mass. This is currently considered in the model allowing for a **Binary Star** sub-class, but a solution based on the chunk sketch from Figure 3 will be kept in mind (though it opens up for a more complex model).

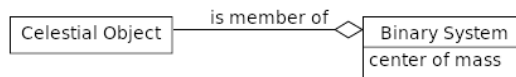


Figure 3. Alternative binary star by composition.

4. Conclusions

This first attempt at modelling exoplanetary systems, to allow discovery and access of this domain's datasets and collection in an interoperable way, will need to widen the community to gather more contributions and requirements, e.g. on details like the class's attributes. Connection to existing modelling efforts within the IVOA has been pointed out and a tighter connection with those specification will be sought.

Acknowledgments. Molinaro, Bignamini, Bonnarel and Louys acknowledge funding by the ASTERICS project, supported by the EU FP H2020 under grant agreement n. 653477.

References

- Benatti, S., Claudi, R., Desidera, S., Gratton, R. G., Lanza, A. F., Micela, G., Pagano, I., Pionto, G., Sozzetti, A., Boccato, C., Cosentino, R., Covino, E., Maggio, A., Molinari, E., Poretti, E., Smareglia, R., & GAPS Team 2016, in *Frontier Research in Astrophysics II*, held 23-28 May, 2016 in Mondello (Palermo), Italy (FRAPWS2016). Online at <https://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=269>, id.69, 69. 1708.04166
- Bonnarel, F., Laurino, O., Lemson, G., Louys, M., Rots, A., Tody, D., & the IVOA Data Model Working Group 2016, IVOA Dataset Metadata Model, version 1.0, IVOA Working Draft. URL <http://www.ivoa.net/documents/DatasetDM/20160317/index.html>
- Lemson, G., Wozniak, H., Bourges, L., Cervino, M., Gheller, C., Gray, N., LePetit, F., Louys, M., Ooghe, B., & Wagner, R. 2012, *Simulation Data Model Version 1.0*, Tech. rep.
- Molinaro, M., Benatti, S., Bignamini, A., & Claudi, R. 2019, in *ADASS XXVII*, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 701
- Murante, G., Maris, M., Palazzi, E., Provenzale, A., Silva, L., Taffoni, G., & Vladilo, G. 2018, in *EGU General Assembly Conference Abstracts*, vol. 20, 6090
- Tody, D., Bonnarel, F., Laurino, O., Louys, M., Rots, A., Ruiz, J. E., Selgado, J., & the IVOA Data Model Working Group 2015, IVOA N-Dimensional Cube Model, version 1.0, IVOA Working Draft. URL <http://www.ivoa.net/documents/NDimCubeDM/20150320/index.html>

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Subaru Telescope Network 5 or STN5 - The New Computer and Network System at the Subaru Telescope

Junichi Noumaru,¹ Thomas Winegar,¹ Eiji Kyono,¹ Hitomi Yamanoi,² and Kiaina Schubert¹

¹*Subaru Telescope, National Astronomical Observatory of Japan, Hilo, Hawaii, U.S.A.; noumaru@naoj.org*

²*Subaru Telescope Mitaka Office, National Astronomical Observatory of Japan, Mitaka, Tokyo, Japan*

Abstract. Subaru Telescope has recently completed the procurement and installation of the fifth contract of the computing environment called Subaru Telescope Network 5 or STN5. Getting ready for in-house management of the next procurement, STN6, was a high priority with STN5. We successfully made the procurement of sufficient computing resources supporting data analysis, instrument control and various service functions in both the Hilo base and the summit facilities.

The analysis environment has been enhanced and the virtual machine environment has been increased at both the Hilo base and the summit facilities. The latter allows for organizational virtual machine (VM)'s developed by divisions within the observatory to be migrated to managed environment.

1. Introduction

Subaru Telescope replaced its core network and computer system called STN5, and started the operation as of March 2018 with a rental contract. The progress as well as the concept of 'rental contract' was reported by Noumaru (2000).

STN5 is the core network and computer infrastructure that interconnects the telescope control system, the astronomical instrument control system, the observation control system called Gen2 (Jeschke & Inagaki (2010)), data archive system in Hawaii (STARS) (Winegar (2008)), various support servers and user computers and devices. STN5 is connected via the dedicated link and via VPN tunnel to the data archive system in Mitaka (MASTARS), another data archive system for general scientists (SMOKA), Japanese Virtual Observatory (JVO) and data analysis and data archive servers for the instrument Hyper Suprime-Cam or HSC that are located in National Astronomical Observatory of Japan in Mitaka, Japan.

2. STN5 configuration

We chose STN5 to follow the basic concept and hardware configuration of the predecessor, STN4, due to enough performance and high stability. As hardware was updated first time for the last five years, performance was improved accordingly. Performance

and software improvement prompted us to deploy virtual machine system so that multiple servers that demand low to medium resource can be consolidated into a single hardware.

Hilo base contains (1) Four (4) load balanced servers for Internal service, DHCP, DNS, LDAP, Web and SSH access, (2) Six (6) VM host servers, (3) Two (2) load balanced DMZ Web servers, (4) Storage access via NFS or CIFS (Samba), (5) Analysis systems, and (6) STARS system composed of three (3) servers w/ 600TB archive for all observational data.

Summit system contains (1) Two (2) servers for DNS, DHCP, LDAP, (2) Four (4) VM Host servers for Telemetry, Analysis and legacy OBCP (Instrument Control Computer) support w/ 100TB usable space, (3) Tape backup, and (4) One (1) Cisco core switch, with a Palo-Alto firewall, VPN Support for remote collaborators.

Mitaka STARS (MASTARS) comprises three (3) servers, providing access to observation data for retrieval.

The STN5 hardware configuration is summarized in Figure 1.

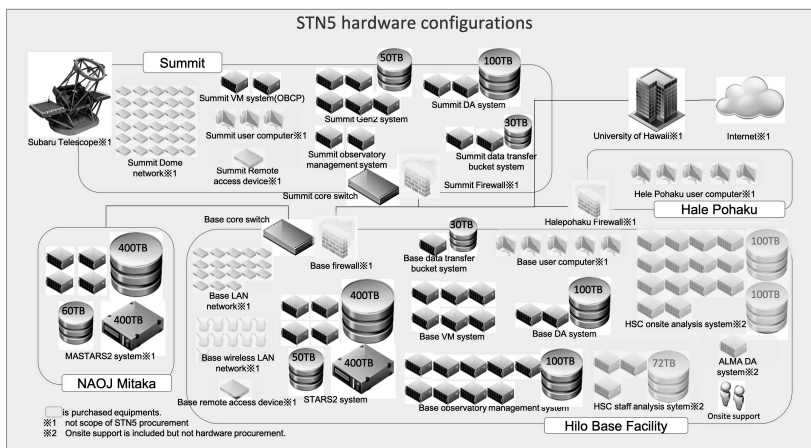


Figure 1. STN5 configurations.

3. Procurement of STN5

Procurement of STN5 was a lengthy process due to the government regulations for the procurement of expensive items as described by Noumaru et al. (2018). Request for Information (RFI) for STN5 was published on the government paper in August 2016. In February 2017, availability of the draft specification and request for comment to the draft specification (RFC) were announced through the government paper.

The specification was revised, made final and made public by June 2017. A winner was identified, and the contract was made in September 2017. Then the detailed design started, and the order of hardware/software was done by the vendor. The entire hardware at Hilo base fits in four 17-inch racks and the hardware at the summit fits in two racks.

4. Network

Subaru Telescope's logical network is complicated due to three domains that it maintains. Subaru Telescope's physical network is also complicated due to diversity of facility locations – three in Hawaii and one in Japan – and due to having backup routes, as shown in Figure 2. Currently, I/O speeds of the firewalls at Hilo base and at the summit facilities toward University of Hawaii (UH)'s network are about 2Gbps and 1Gbps respectively. This limits the available bandwidth to the Internet via UH's network although the available network speed is 10Gbps. Most services outside of our network are provided with HTTP/HTTPS. Registered staff and collaborators can access internal resources with SSL VPN.

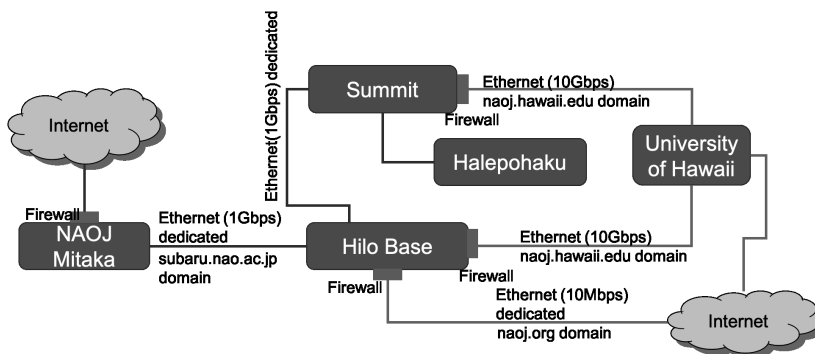


Figure 2. WAN configurations.

5. Summit VM subsystem

Summit VM subsystem comprises four host servers with 6TB local disk space, 2 x 10Gbps NIC and 32-core CPUs and supports five VLAN sub-interfaces for legacy support on older OBCPs that were migrated to VM. As shown in Figure 3, Host Server 1 serves a primary Fiber Channel connection to backend storage. Host Server 2 serves a secondary Fiber Channel connection to storage. VM images are stored on RAID storage, with Host Server 1 acting as NFS server to other servers. Any VM can be started on any Host Server. If Host Server 1 becomes unavailable, automate script will configure RAID and Host Server 2 to become active, and to serve as NFS of VM images and storage to allow observation to continue after minimal downtime.

6. Data archive system (STARS) FITS archive and query

We are developing a FITS correction system for modification of FITS keywords calculated by post-observation reduction clusters. Example keywords are for seeing and transparency. Users can download either FITS originals with ASCII corrections or the updated FITS corrected file. Offsite observers now may download FITS files approximately within 30 minutes after observation for the download speed of up to 30 GB/hour.

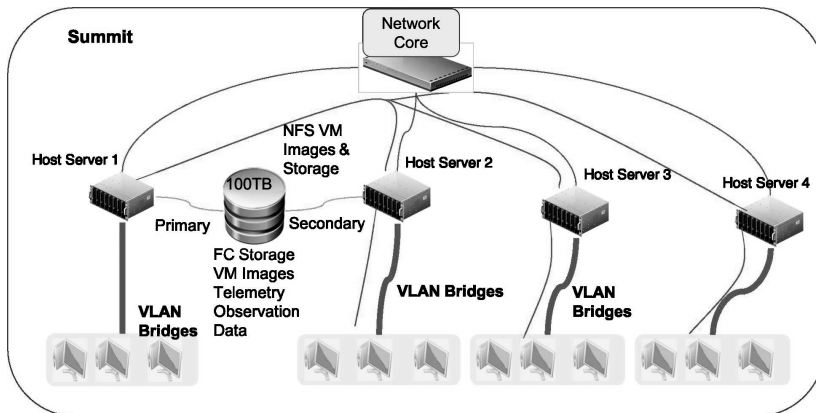


Figure 3. VM system at the summit.

7. Summary

STN5, the computer and network infrastructure for Subaru Telescope, was procured with a rental contract through lengthy 19-month of process and the contract will continue through 2023. The goal of STN5 is to allow for Computer & Data Management Division to be relieved of system and network administration during the first year of the contract, and to be able to get ready for the next computer and network system - STN6 - that will take over STN5 from 2023 and that will be mostly managed by our employees.

Hardware was upgraded and the basic concept and configuration of STN5 were chosen to be quite similar to those for STN4. In STN5, we consolidated servers for various services into virtual machines. For system redundancy, RAID 6 storage is connected to two host servers. The VM system now includes data analysis servers which demand much CPU and I/O.

References

- Jeschke, E., & Inagaki, T. 2010, in *Software and Cyberinfrastructure for Astronomy*, vol. 7740 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 77400S
- Noumaru, J. 2000, in *Observatory Operations to Optimize Scientific Return II*, edited by P. J. Quinn, vol. 4010 of *Proceedings of SPIE*, 10
- Noumaru, J., Winegar, T., Kyono, E., Yamanoi, H., & Schubert, K. 2018, in *ADASS 2016 Proceedings*, *ASP Conference Series* (in press)
- Winegar, T. 2008, in *Observatory Operations: Strategies, Processes, and Systems II*, vol. 7016 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 70160M

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Data Products from the Europa Imaging System (EIS) on Europa Clipper

G. Wesley Patterson,¹ Alfred S. McEwen,² Elizabeth P. Turtle,¹
Carolyn M. Ernst,¹ and Randolph L. Kirk³

¹*Johns Hopkins Applied Physics Laboratory, Laurel, MD, USA*

²*Lunar and Planetary Lab, University of Arizona, Tucson, AZ, USA;*
mcewen@lpl.arizona.edu

³*Astrogeology Science Center, U.S. Geological Survey, Flagstaff, AZ, USA*

Abstract. The Europa Imaging System (EIS) consists of narrow- and wide-angle cameras; each can image in framing, pushbroom (including color), and stereo modes. Anticipated data products include single-image formats, mosaics, and digital terrain models.

1. Introduction

NASA's Europa Clipper mission will investigate Europa's potential habitability via study of its subsurface water, ice shell, surface, and exosphere (Pappalardo et al. 2017). The Europa Imaging System (EIS) (Turtle et al. 2016) comprises two of 10 science instruments. A suite of EIS datasets will be produced to: Constrain formation processes of landforms; identify relationships to subsurface structures detected by ice-penetrating radar (Blankenship et al. 2018); search for evidence of recent or current activity; constrain ice-shell thickness from Europa's shape and topography; and characterize the surface at meter scales and identify potential future landing sites.

The Europa Clipper mission will orbit Jupiter and perform over 40 close flybys of Europa (<100 km altitude). EIS consists of narrow-angle (NAC) and wide-angle (WAC) cameras, both of which can operate either in pushbroom mode with up to 6 colors or in framing mode in a clear bandpass. The flyby observation geometry leads to constantly changing range (controlling spatial resolution) and photometric angles (illumination, viewing, and phase angles), complicating production of uniform mapping products. The NAC 2-axis gimbal enables nearly global (~90%) clear-bandpass coverage at <100 m pixel scale, and regularly scheduled spacecraft scans at ~30,000-km altitude will provide global color coverage at ~300 m/pixel. These global datasets provide context for high-resolution (0.5–25 m/pixel) images and local mosaics. Existing imaging coverage of Europa from the Voyager and Galileo missions is ~20 km/pixel at the global scale down to 6 m/pixel in just a handful of images; EIS will improve on this by a factor of ~10³. Each flyby of Europa will produce a characteristic series of observations as a function of range (Table 1).

Table 1. Typical orbital set of EIS observations

Range to Europa (km)	Observations
≤1,000,000	NAC framing plume search and monitoring at ≤10 km/pixel
≤100,000	NAC framing limb profiles at ≤1 km/pixel for global shape
≤40,000	NAC pushbroom full-disk color at 250–400 m/pixel
5,000–10,000	NAC framing to build near-global map at 50–100 m/pixel
≤4,600	WAC framing limb profiles at ≤1 km/pixel for global shape
2,000–5,000	NAC framing regional mosaics and stereo at 20–50 m/pixel
25–2,500	NAC and WAC pushbroom color and stereo at high resolution

2. Data Products

EIS data need significant processing to best achieve the science objectives. The main data products fall into two categories: automated pipeline processing that is completed rapidly once the data are available (Table 2), and special products that require extra time and effort (Table 3). All products will be generated directly in Planetary Data System (PDS-4) formats (Hughes et al. 2018).

Table 2. EIS Standard (Pipeline) Data Products

PDS-4 Product	Description
Raw Images	Uncompressed images with PDS-4 labels
Partially Processed	Radiometric calibration applied to raw images, but no pixel re-sampling (preserves full resolution)
Calibrated	Radiometric and standard geometric calibrations applied, map projected co-registered colors
Derived	Digital Terrain Models (DTMs) from WAC 3-line stereo
Browse	Reduced-scale jpeg images, 3-color images, stereo anaglyphs, and WAC color flyover movies

Standard products (Table 2) will be produced by automated software that pulls data products from the Mission Operations Center and triggers processing as soon as the data for a particular product are complete. Most of the pipeline processing procedures are mature and used by other projects (e.g., McEwen et al. 2010). Much of the data processing will rely on the USGS ISIS3 software package (Sides et al. 2017). "Raw" image format is straightforward for framing images. Individual pushbroom images are typically segments of longer continuous images. The line readout time must be re-calculated at intervals to match the rate of motion of the ground when using digital time delay integration (TDI), to avoid smearing. NAC pushbroom images will be relatively short images and can be map projected at a uniform scale. However, the WAC pushbroom images will be very long images acquired continuously along the flyby ground-track, with pixel scale varying from as little as ~5 m up to ~500 m. Map-projected versions of the entire strip would be enormous files, contain mostly empty space, and be highly oversampled at their (high-altitude) ends. Thus, we plan to divide each strip into sections in which raw scales vary by 2x, with no more than 2x oversampling. These scales will be 5 m/pixel for raw data at ≤10 m/pixel, 10 m/pixel for raw data at 10–20 m/pixel, and so on for 20, 40, 80, 160, and 320 m/pixel. We plan to use the oblique cylindrical projection because it tracks a flyby strip regardless of where it is

on Europa. We will also reproject all images (with the same 2x scale segmentation) to a projection such as equirectangular to mosaic data from multiple orbits. The Galilean satellites are map projected over triaxial ellipsoid shape models, and the Europa Clipper project has chosen to use positive longitude east and west (both described in labels); when describing longitudes it is essential to add "E" or "W" to avoid confusion.

Table 3. EIS Special Data Products

Derived Product	Description
NAC Global Color	Best low-phase coverage in 7 bandpasses from full-disk scans, photometrically normalized and mosaicked in a standard map format.
Global Panchromatic Mosaic	Best coverage via clear bandpass, photometrically normalized and mosaicked at 100 m/pixel in standard map format.
Regional Panchromatic Mosaics	Regional mosaics at 10–100 m/pixel from sets of clear-bandpass images with uniform lighting angles.
NAC DTMs	Sets of NAC images with similar lighting and different viewing angles, processed into DTMs at ~ 4x the worst image scale.
Updated C-kernels	Output from pushbroom jitter correction algorithm (see text).
Bond Albedo Maps	Derived from NAC color scans plus near-IR data (Blaney et al. 2017).
Global Shape Dataset	Derived from ~1 km/pixel limb profiles, format TBD.
Geodesy Dataset	Precision control point network tied to radar altimetry (Blankenship et al. 2018).
Browse Products	Merged color and monochrome mosaics, NAC stereo anaglyphs, 3-color products, reduced scale products.

Special products (Table 3) are expected to require personal attention and will not be produced as rapidly as pipeline products. EIS images in both pushbroom and framing modes will be susceptible to geometric distortions from pointing jitter, which will be measured and corrected. In pushbroom mode, image lines covering the same surface features at slightly different times will be used to model the absolute pointing to ~1 pixel accuracy (Sutton et al. 2018). With an all-digital camera we can select the spacing (timing) of lines to best measure the typical spacecraft jitter frequencies seen in flight. We expect to nearly always acquire both color and stereo images with the WAC, providing ample data for jitter measurements, whereas additional narrow (~100 pixel wide) NAC images may be returned for additional jitter frequencies. Framing images are acquired via rolling shutter readout taking at least 26 ms, so very high-frequency jitter could cause geometric distortions, likely to be less than 1 pixel over a frame, but nevertheless important to precision geodesy measurements. Thus a new checkline readout capability has been developed to correct frames to 1/10 pixel accuracy (Kirk et al. 2018). The products of the pushbroom jitter models are updated camera pointing angles to be archived as PDS-4 SPICE products, and for framing images the information will be stored in the image labels. These corrections are then used for subsequent geometric reprojections and DTM production.

Photometric functions must be derived and applied to EIS products to generate uniform mosaics, quantify surface changes due to current activity, and derive maps of

Bond albedo. A variety of functions such as those of Hapke and Kaasalainen-Shkuratov have been used to describe image brightness as a function of photometric angles (e.g., Domingue et al. 2016). These models have a set of unknown parameters that must be fit to the calibrated observations in each spectral band, and may vary from place to place across Europa. The full-disk NAC scans at ~ 300 m/pixel will be ideal to derive these fits for each surface area of ~ 1 km², as there will be ~ 30 overlapping observations with a broad range of photometric angles. The Bond albedo accounts for all of the light scattered from a body at all wavelengths and scattering angles, and is a necessary quantity for determining how much solar energy a surface absorbs. The resulting prediction of re-radiated solar heat can be subtracted from thermal-IR observations to measure any excess heat that is generated endogenically, such as in (cryo-)volcanically active regions. The NAC dataset will be excellent to derive this quantity from 350–1050 nm, but $\sim 30\%$ of the sun's energy radiates at longer wavelengths. Data from the Mapping Imaging Spectrometer for Europa (MISE) (Blaney et al. 2017) will fill the spectral gap.

Browse products (in PDS-4 terminology) will include excellent public outreach products such as 3-color images and mosaics, stereo anaglyphs, and color and stereo-anaglyph flyover movies. Although current planning is focused on Europa observations, with other targets considered for calibration opportunities, it is likely that observations of Jupiter, its faint rings, and satellites will be possible; we will produce at least a set of standard data products for any such targets. We plan to make some of the images available on the mission website as soon as they're available, in addition to formal archiving.

References

- Blaney, D. L., et al. 2017, in Lunar and Planetary Science Conference, vol. 48 of Lunar and Planetary Science Conference, 2244
- Blankenship, D., et al. 2018, in 42nd COSPAR Scientific Assembly, vol. 42 of COSPAR Meeting, B5.3
- Domingue, D. L., Denevi, B. W., Murchie, S. L., & Hash, C. D. 2016, *Icarus*, 268, 172
- Hughes, J. S., et al. 2018, *Planetary and Space Science*, 150, 43
- Kirk, R. L., Shepherd, M., & Sides, S. C. 2018, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-3, 735
- McEwen, A. S., et al. 2010, *Icarus*, 205, 2
- Pappalardo, R. T., Senske, D. A., Korth, H., Klima, R., Vance, S. D., & Craft, K. 2017, *European Planetary Science Congress*, 11, EPSC2017-304
- Sides, S. C., et al. 2017, in Lunar and Planetary Science Conference, vol. 48 of Lunar and Planetary Science Conference, 2739
- Sutton, S., et al. 2018, in *Planetary Remote Sensing and Mapping*, Taylor and Francis Group/CRC Press, edited by B. Wu, K. Di, J. Oberst, & I. Karachevtseva, 91
- Turtle, E. P., McEwen, A. S., Osterman, S. N., Boldt, J. D., Strohbehn, K., & EIS Science Team 2016, in 3rd International Workshop on Instrumentation for Planetary Mission, vol. 1980 of LPI Contributions, 4091

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

The CDS HEALPix Library

Francois-Xavier Pineau and Pierre Fernique

*Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg,
UMR 7550, F-67000 Strasbourg, France;
francois-xavier.pineau@astro.unistra.fr*

Abstract. The CDS is releasing a new HEALPix library implemented in Java, Rust and WebAssembly. The library focuses on the CDS needs, on performances and accuracy and is distributed under the 3-clause BSD license. Aladin desktop has already started to integrate the Java version of the library in its code. The current state of the library and its specific features are presented in this article.

1. Motivations

The motivations bringing the CDS to develop an HEALPix library from scratch are diverse. First of all, we wanted to develop an internal expertise in a key component of both CDS services and the HiPS IVOA standard. It will allow us to improve the HEALPix support in Aladin Lite by bringing the deepest addressable resolution from order 13 to order 24 and to add support for polygons. We also wanted to control the license to be able to switch from the GPL (“official library”) to the 3-clause BSD license (“CDS library”). The 3-clause BSD will allow us to change the Aladin Lite license and to be compatible with, for example, the Astropy one. Finally, by writing our own library, making changes fitting with the CDS needs (mainly in Aladin, Aladin Lite and the cross-match service) will be easier.

2. Languages

We considered different programming languages for this library: Java, which is widely used at CDS; Rust, which we wanted to test and has a good support for WebAssembly; and C, which is often supported by large projects for the development of external modules.

2.1. Java

Since it is a popular language widely used at CDS – Aladin, SIMBAD and the cross-match service are fully written in Java – the HEALPix library has been developed first in Java. All features mentioned in this document were made available in the Java version of the library.

2.2. Rust and WebAssembly

Rust is a recent and promising open-source language sponsored by Mozilla which *pursues the trifecta: safety, concurrency, and speed* (Rust weekly newsletter). It has the additional aim to offer *high-level ergonomics and low-level control* (online Rust book). Since it is a compiled language, we use Rust to generate both WebAssembly files and static or dynamic libraries that can be called from Python or PostgreSQL. So far, mainly basic HEALPix features meeting with the Aladin Lite needs have been implemented in Rust (cell number from coordinates, cell center, cell vertices, cell neighbors, approximate cells-in-cone, projection/de-projection).

WebAssembly is a bytecode standardized by the W3C and compatible with all recent Web browsers. It aims to complement Javascript by providing better performance, and can be generated from compiled languages like C, C++ or Rust. We mainly target WebAssembly for Aladin Lite, but the library may also meet with the needs of other web applications using HEALPix.

2.3. The C programming language?

Rust pre-compiled binaries are similar to C and could be distributed in software like Astropy. However, installing Rust tools is necessary if a user wants to manually compile a module from the source code. Integrating Rust code into large projects like Astropy or PostgreSQL modules is thus not straightforward. This issue may bring us to develop a C version of the library.

3. Features

The features implemented in the HEALPix library meet the specific needs of CDS tools. So far, the code does not support the RING scheme – but it offers functions converting a NESTED cell number into a RING cell number and vice-versa – and Fast Fourier Transforms which are extensively used in the cosmology community. It does support a number of interesting features beyond a basic set. We mention here but a few.

3.1. BMOC for cone and polygon queries

A BMOC is an extension of a MOC storing for each cell an additional status flag telling if the cell is either partially or fully covered by the area the MOC represents. This binary coverage information allows avoiding useless time-consuming distance computations when performing cone-search queries on a table: the sources inside a cell “fully” covered by the cone does not have to be tested.

The library offers an approximated and an exact “cells-overlapped-by-cone” solution. In both cases the result is a BMOC. The exact solution does not contain false positive cells and it thus avoids possibly useless distance computations and disk accesses.

In addition, the library offers a very-fast – but with large approximations – cells-overlapped-by-cone function dedicated to map/reduce based cross-matches.

The library also supports self intersecting polygons of any size providing a BMOC as a result. The algorithm so far resorts to an approximation: it considers a cell border between two vertices as if it was a great-circle arc. Although an exact solution is possible it would be computationally less efficient.

3.2. Other features

The library supports an internal projection/de-projection in addition to a version compatible with the WCS HPX projection. We recall that a projection/de-projection consists in computing Euclidean from spherical coordinates and vice-versa.

Possibly useful for cross-match applications the library also provides: a function computing an upper limit on the largest center-to-vertex distance depending both on the order and on the position of the cell on the sky, and an ordered list of small cells surrounding a larger cell to take into account border effects while ensuring the sequential access to data stored on spinning disks, etc.

4. Technical details

We provide in this section a few key mathematical elements of the library internals.

4.1. Projection: simplified equations

HEALPix is quite extensively described in Calabretta (2004), Górski et al. (2005), Calabretta & Roukema (2007) and Reinecke & Hivon (2015). It is first of all an equal-area projection composed from two other projections. Internally we have chosen a projection scale such that all coordinates in the projection plane are $\in [0, 8[$ on the X -axis and $\in [-2, 2]$ on the Y -axis. The internal simplified equations are:

Cylindrical equal-area projection in the equatorial region:

$$\begin{cases} X &= \alpha \times \frac{4}{\pi} \\ Y &= \sin(\delta) \times \frac{3}{2} \end{cases} \Rightarrow \begin{cases} \alpha \in [0, 2\pi] & \rightsquigarrow X \in [0, 8] \\ \sin \delta \in [-\frac{2}{3}, \frac{2}{3}] & \rightsquigarrow Y \in [-1, 1] \end{cases}$$

Collignon (pseudo-cylindrical equal-area) projection in the polar caps, for $\alpha \in [0, \pi/2]$ and $\sin \delta > 3/2$:

$$\begin{cases} t &= \sqrt{3(1 - \sin \delta)} \\ X &= (\alpha^{\frac{4}{\pi}} - 1)t + 1 \\ Y &= 2 - t \end{cases} \Rightarrow \begin{cases} \alpha \in [0, \frac{\pi}{2}] & \rightsquigarrow t \in [0, 1] \\ \sin \delta \in [\frac{2}{3}, 1] & \rightsquigarrow \begin{matrix} X \in]0, 2[\\ Y \in]1, 2] \end{matrix} \end{cases}$$

4.2. Precision at poles

The formula $t = \sqrt{3(1 - \sin \delta)}$ causes non-negligible numerical inaccuracies near the poles due to the $1 - \sin \delta$ expression it contains: in facts, $\arcsin(1 - 1.0 \times 10^{-15}) \approx 89.99999919$ deg, and $\frac{\pi}{2} - \arcsin(1 - 1.0 \times 10^{-15}) \approx 2.917$ mas.

We thus replaced the previous equation by the equivalent but numerically stable form: $t = \sqrt{6} \cos(\frac{\delta}{2} + \frac{\pi}{4})$. This form is also computationally less expensive since we spare a time-consuming square-root operation (the square-root applying here on a constant instead of a variable).

4.3. The exact cells-in-cone solution

Basically, if the four vertices of a cell are inside a cone then the cell is fully overlapped by the cone. If at least one vertex is inside the cone and one vertex is outside then

the cell is partially overlapped by the cone, but the cone also contains 4 special points such that the slope of the tangent line to the projected cone on that point equals plus or minus one. If a cell contains such a point then it is also partially overlapped by the cone. Finally, for large cells containing no vertices and no special points, we have to test if the center of the cone is inside the cell.

To compute the coordinates of the four “special” points, we first use the Haversine formula to get an accurate cone expression at small radii:

$$\Delta\alpha = 2 \arcsin \left(\sqrt{\frac{\sin^2 \frac{\theta}{2} - \sin^2 \frac{\delta - \delta_0}{2}}{\cos \delta_0 \cos \delta}} \right).$$

In the equatorial region, the equation of tangent lines is:

$$\frac{d\Delta X}{dY} = \frac{d\Delta X}{d\Delta\alpha} \frac{d\Delta\alpha}{d\delta} \frac{d\delta}{dz} \frac{dz}{dY} = \pm 1.$$

The projection formulae $z = \sin \delta$, $X = 4/\pi\alpha$, $Y = 3/2z$ lead to $\frac{d\Delta X}{d\Delta\alpha} = 4/\pi$, $\frac{d\delta}{dz} = \frac{1}{\cos \delta}$, $\frac{dz}{dY} = 2/3$ and, finally, we find the special points latitudes by solving numerically:

$$\frac{1}{\cos \delta} \frac{d\Delta\alpha(\delta)}{d\delta} \mp \frac{3\pi}{8} = 0.$$

In the polar caps, the equation of tangent lines is:

$$\frac{d\Delta X}{dY} = \frac{d\Delta X}{d\delta} \frac{d\delta}{dz} \frac{dz}{dY} = \pm 1$$

with $z = \sin \delta$, $t = \sqrt{3(1-z)} = \sqrt{6} \cos(\frac{\delta}{2} + \frac{\pi}{4})$, $X = (\frac{4}{\pi}\alpha - 1)t + 1$, $Y = 2 - t$, leading to $\frac{d\delta}{dz} = \frac{1}{\cos \delta}$, $\frac{dz}{dY} = \frac{2}{3}t$ and, finally, we find the special points latitudes by solving numerically:

$$\frac{t(\delta)}{\cos \delta} \frac{d}{d\delta} \left[\left(\frac{4}{\pi}\alpha(\delta) - 1 \right) t(\delta) \right] \mp \frac{3}{2} = 0.$$

5. Conclusion

The CDS has been developing a new HEALPix library which has a permissive license and contains innovative features like the BMOC. Most of the features have been tested and the new Java library is been replacing the “standard” one in Aladin. The library is publicly available on GitHub.

References

- Calabretta, M. R. 2004, ArXiv Astrophysics e-prints. astro-ph/0412607
 Calabretta, M. R., & Roukema, B. F. 2007, MNRAS, 381, 865
 Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. 2005, ApJ, 622, 759. astro-ph/0409513
 Reinecke, M., & Hivon, E. 2015, A&A, 580, A132. 1505.04632

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Availability of Hyperlinked Resources in Astrophysics Papers

P. Wesley Ryan,¹ Alice Allen,^{1,2} and Peter Teuben²

¹*Astrophysics Source Code Library; wes@ascl.net*

²*Astronomy Department, University of Maryland, College Park*

Abstract. Astrophysics papers often rely on software which may or may not be available, and URLs are often used as proxy citations for software and data. We extracted all URLs from two journals' 2015 research articles, removed those from certain long-term reliable domains, and tested the remainder to determine what percentage of these URLs were accessible in October 2018.

1. Introduction

Astrophysics, like most disciplines, relies on software and repositories of data for its research. Software source codes and raw data are often too large and complex to share within the papers in which they are used. Although standards for citing software, such as the IDs assigned by the Astrophysics Source Code Library (ASCL)¹, have recently been developed and are increasingly used, they have yet to reach universal adoption. As a result, information relevant to a paper may be accessible from that paper only via a URL. Howison & Bullard (2015) found that 5% of software mentions in a sample of biology papers referenced the software only by a link in the text. (In comparison, 31% used only an in-text mention of the name of the software.) Even when software is properly cited, it may be available only as a binary or web tool, or not available at all.

Although some pre-WWW hypertext system designs provided for guaranteed resource persistence, the Web itself does not; as a result, URLs may fail to resolve to the proper resource. Websites are reorganized, graduate students move on, and professors retire. In each case, links may break, including links contained in published papers.

Resource accessibility is important to research transparency, repeatability, and reproducibility. If the code used to compute a result is not available, it cannot be audited for bugs; if the data upon which a computation is carried out is inaccessible, it cannot be reanalyzed. Because of the importance of resource accessibility, and because these resources may be referred to with hyperlinks, we have undertaken a research project dedicated to studying the availability of hyperlinked resources over time.

Similar studies have been carried out in other fields. Mangul et al. (2018) found that 24% of URLs in a large sample of biomedical papers published from 2000 to 2017 were broken, and 4% more were unreachable due to connection timeouts.

¹<https://ascl.net>

2. Context and methods

The present paper is a follow-up to a 2018 study conducted by the authors (Allen et al. (2018)). As part of that study, we extracted the HTTP(S) and FTP hyperlinks from all papers published in *Astronomy & Astrophysics (A&A)* in 2015, excluded links to nine commonly-referenced ‘infrastructure’ sites that we knew to be available, tested the rest for availability, and assigned them to one of three categories: consistently available, consistently unavailable, and inconsistently available. For HTTP(S) links, we used an automated checker derived from one in use at the Astrophysics Source Code Library, and defined consistently available links as those which returned the 200 OK status code every time we tested them; consistently unavailable links as those which returned other status codes, contained domain names which could not be resolved, or had other errors²; and inconsistently available links as those which sometimes returned 200 OK status codes and sometimes did not. Links that were inconsistent in the status codes they returned but never returned a 200 OK status code were categorized as consistently unavailable.

Due to the small number of FTP links contained in our dataset and the limitations of the ASCL link checker, we opted in our original research to check these links by hand, and assign them to our categories by whether or not they resolved to a resource.

In our follow-up research, we tested the same HTTP(S) links one year after our initial checks (which were carried out in September and October 2017), and extracted and tested the HTTP(S) links in the papers published in the *Astrophysical Journal (ApJ)* in 2015, using the LExTeS package (Ryan 2017) that we developed for the initial paper, with minor improvements.³ In this paper, we consider only HTTP(S) links; since the *A&A* dataset contains only 30 FTP links and the *ApJ* dataset contains only 45, the effect of discarding them is negligible.

3. Results

Table 1 shows the percentage of links by availability category. Of the 2,528 HTTP(S) links in the *A&A* dataset, 2,176 were consistently available, 322 were consistently unavailable (4 of which were consistently unavailable but for inconsistent reasons, so are not listed in Table 2), and 30 were inconsistently available. Of the 3,141 HTTP(S) links in the *ApJ* dataset, 2,626 were consistently available, 460 were consistently unavailable (6 of which were consistently unavailable but for inconsistent reasons, so are not listed in Table 2), and 55 were inconsistently available.

²Failed domain lookups and other errors that do not correspond to HTTP status codes were assigned codes with negative numbers.

³The original version of LExTeS was written in Python 2 and designed to be run on Linux. Since the present author now has Python 3 and Windows, the software was ported from Python 2 to Python 3 and certain Linux-specific idioms, such as the expectation of Unix-style shell glob expansion, were replaced with more portable ones. In addition, the now-deprecated library pyPdf, used for extracting hyperlinks from PDFs, was replaced with PyPdf2.

Table 1. Percentage of links by category

	Up	Down	Inconsistent
A&A (Sep./Oct. 2017)	86.8%	10.6%	2.6%
A&A (Oct. 2018)	86.1%	12.7%	1.2%
ApJ (Oct. 2018)	83.6%	14.6%	1.8%

Table 2. Consistently unavailable links that always returned the same error code

Error code	A&A (Oct. 2018)	ApJ (Oct. 2018)
-7 Timeout error	0	1
-6 Connection reset	0	2
-5 Value error	0	9
-4 Bad status line	1	0
-3 Socket error	2	0
-2 SSL certificate error	6	0
-1 Lookup failed	97	172
302 Found	0	0
400 Bad request	0	2
401 Unauthorized	3	7
403 Forbidden	34	32
404 Not found	167	220
500 Internal server error	5	5
502 Bad gateway	1	1
503 Service unavailable	2	3

4. Conclusions and Future Work

The rate of link decay seems not to be constant over time. Although a small number of the links in the A&A dataset that were categorized as consistently unavailable in 2017 recorded some successful checks in 2018, the overall percentage of broken links shows a lower per-year decrease from 2017 to 2018 than from 2015 to 2017.

We plan to test our current datasets of URLs periodically and build additional datasets of links from other years and other journals. Our goal is to track link health and the degree of reliance on possibly ephemeral methods of referencing code and data, and investigate whether recent and continuing changes in citation methods improve the overall availability of these resources going forward.

References

Allen, A., Teuben, P. J., & Ryan, P. W. 2018, The Astrophysical Journal Supplement Series, 236, 10. URL <http://stacks.iop.org/0067-0049/236/i=1/a=10>
Howison, J., & Bullard, J. 2015, Journal of the Association for Information Science and Technology, 67, 2137. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23538>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23538>

616

Ryan, Allen, and Teuben

Mangul, S., Mosqueiro, T., Duong, D., Mitchell, K., Sarwal, V., Hill, B., Brito, J., Littman, R., Statz, B., Lam, A., Dayama, G., Grieneisen, L., Martin, L., Flint, J., Eskin, E., & Blekhman, R. 2018, bioRxiv. <https://www.biorxiv.org/content/early/2018/10/25/452532.full.pdf>, URL <https://www.biorxiv.org/content/early/2018/10/25/452532>

Ryan, P. W. 2017, LExTeS: Link Extraction and Testing Suite, Astrophysics Source Code Library. 1711.018



Alice Allen and Wesley Ryan at the registration desk (Photo: Peter Teuben)



A surprise snow “storm” on the last day of conference (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Data Processing of the Stratospheric Terahertz Observatory-2 [CII] Survey

Russell Shipman,^{1,2} Youngmin Seo,³ Volker Tolls,⁴ William Peters,⁵ Ümit Kavak,^{2,1} Craig Kulesa,⁵ and Chris Walker⁵

¹*SRON, Groningen, Groningen, The Netherlands; R.F.Shipman@sron.nl*

²*Kapteyn Astronomical Institute, Groningen, The Netherlands*

³*JPL Caltech, Pasadena, California, USA*

⁴*Harvard-Smithsonian Center for Astrophysics, Cambridge, Massachusetts, USA*

⁵*University of Arizona, Tuscon, Arizona, USA*

Abstract. The second flight of the Stratospheric Terahertz Observatory (STO-2) was a balloon mission to survey parts of the Galactic Plane at [CII] transition at 1.9 THz. STO-2 surveyed approximately 2.5 deg^2 of the Galactic Plane at a spatial resolution of $1'$. The STO-2 data suffer significant system drifts that are only partially addressed by the observing cadence. A slightly altered calibration scheme is presented to address these drifts. We show how it was possible to extract calibrated data from STO-2 scans and, based on the work presented here, make recommendations for the future GUSTO mission.

1. Introduction

Observing abundant atoms and ions traces the dynamics and life cycle of the interstellar medium (ISM) in galaxies. Singly ionized carbon, [CII], is particularly useful to trace the ISM. With an ionization potential of 11.26eV, carbon is easily ionized by the UV radiation from young hot stars. High spatial and spectral observations of [CII] show not only where the ion is but how it is moving. Sparse spatial but velocity resolved [CII] emission has recently been studied along 500 lines-of-sight throughout our Galaxy (Langer et al. 2014) and provided significant insights into the internal workings of the ISM in our Galaxy.

The Stratospheric Terahertz Observatory is a double sideband (DBS) heterodyne spectrometer with the goal of surveying parts of the Galactic Plane at the [CII] 1.9THz transition at a spectral resolution of 0.16 km/s. To minimize the influence of Earth's atmosphere, STO was put on a high altitude balloon platform which flew at 40 km above the South Pole. The second flight of STO, STO-2, obtained data from December 15 to 30, 2016. In that time more than 300,000 fully sampled scans of the Galactic Plane were made. Details of the STO experiment can be found in Walker et al. (2010).

2. Observations

For the [CII] survey, STO-2 made use of the On-The-Fly mapping (OTF) mode. OTF is a means of mapping a region by continuously scanning and intermittently reading out a detector (Mangum et al. 2000). OTF is a highly efficient means of covering a large region of the sky with single detectors or small arrays of detectors. The OTF technique uses the standard vane calibration of radio telescopes (Kutner & Ulich 1981) which combines data of a sky reference position free of emission as well as data on an internal load of known temperature. STO-2 has an internal hot load. The main constraint in OTF mapping is the timing between readouts of the instrument during the scan (ON), the time between load measurements (HOT) and the time between the sky measurements (REF).

An OTF scan is shown in Figure 1. The scan begins with a sky (REF 1) observation along with a hot load (HOT). The telescope points to the beginning of the mapping region and repeats the load measurement (HOT_B). Then starts integrating on the sky while moving along the scan leg. The integrations are readout frequently to minimize source blurring. At appropriate intervals and at the end of the scan the internal hot load is observed (HOT_E). This pattern continues on a new scan parallel but offset by a fraction of the spatial resolution of the instrument.

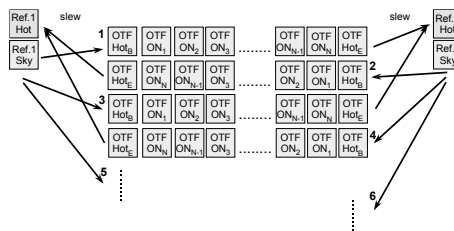


Figure 1. Observation sequence of an OTF scan. The subscripts represent increasing readouts along a scan

As can be seen in the Figure 1, the timing between successive ON readouts is the fastest, followed by the intermittent HOT, then by the REF measurements. This pattern comprises the observation cadence.

2.1. Standard calibration

The reference observations (REF) and repeated hot measurements (HOT) are combined to calibrate the system onto a known radiometric scale (Kutner & Ulich 1981). Reference positions should be emission free and the system should be stable. The calibration of a DBS receiver is give by:

$$T_A^* = T_{sys} \frac{(ON - REF)}{REF} \quad (1)$$

$$\text{with, } T_{sys} = 2 \times \frac{T_{HOT} - Y \times T_{REF}}{Y - 1} \text{ with } Y = \frac{HOT}{REF}$$

T_{sys} is the system noise temperature. The Y factor is the ratio of the raw HOT counts to the raw REF counts. T_{HOT} is the hot load temperature (290K) while T_{REF} is the effective temperature of the blank sky at 1.9 THz (45K). The values of the Y factor

and the REF measurement per channel are linearly interpolated to the time of the ON readout.

2.2. Radiometric noise and drift noise

Radio observations are afflicted with mainly two different noise types: radiometric (white) noise and drift noise. White noise is independent of frequency of observing and can be reduced by longer integrations. Drift noise increases over longer time periods and in general cannot be reduced by longer integrations or repeated measurements.

Drift noise originates from the instability of the detector system. Understanding the timing of instabilities is needed and a proper observation sequence must be chosen to minimize drift effects. Whereas white noise is flat across spectral channels (bandpass), drifts result in spectra which fluctuate over the bandpass. The resulting spectra suffer from poor baselines and/or standing waves. Poor baselines limit the useful information present in the signal by confusing spectral features of the sky with drift noise.

The right hand side of Figure 2 shows calibrated scan spectra for one leg of an OTF scan. The drift noise has built up spectra "features" which repeat over many scans.

2.3. Addressing drift noise

The calibration requires stability of the entire system. The reference observations are usually taken a significant time before and after the OTF scan. The drift time constant is described by the Allan time (Allan 1966) and observations should be designed with this drift time in mind.

The frequency of the load observations (HOT) helps make up the overall cadence. Another component is the reference observation. Often, the stability time is short compared to the cadence of the reference measurements. This implies that the references observations, although necessary for calibration, may leave larger than desired drifts. The hot load can be used to address the system drift since changes in the HOT reflect the changes of the system much closer in time. In other words, an intermittent load scan can be used to help stabilize the calibration.

To account for drifts and better use the system monitoring aspect of frequent load measurements the calibration equation can be altered to create an "interpolated" REF signal.

$$T_A^* = T_{sys} \frac{(ON - intREF)}{intREF} \quad (2)$$

In this case, $intREF = HOT(t) \times \frac{REF(t_0)}{HOT(t_0)}$ and is linearly interpolated to the scan readout time t . t_0 is the time of the reference scan and accompanying hot.

Interpolating between standards is not new and a full discussion for OTF observations is given in Ossenkopf (2009). In the presence of significant system drifts, all calibration factors need to be interpolated in time to match the scan integration time. This can be seen in the right hand side of Figure 2 where a significant improvement is gained by normalizing again by the HOT scan closest to the OTF integration.

Even after altering the calibration equations, often ripples are still present in the baseline. Common methods exist to address poor baselines including fitting a low order polynomial or even sine waves if the pattern is periodic. Such baseline fits might be impractical on survey data since they require knowing which velocities should be emission free. Since the drifts build up over many scans, perhaps machine learning

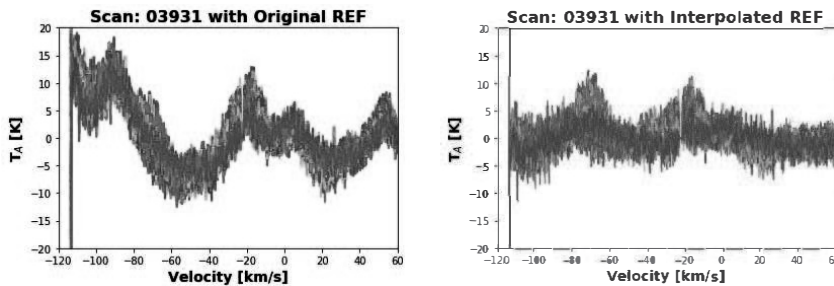


Figure 2. *Left:* An OTF scan with original calibration (Equ. 1) applied to a series of readouts along a scan. *Right:* The same scan after applying the HOT scan stabilized calibration (Equ. 2).

techniques can be adopted to address these drifts. Nevertheless, given that a careful application of calibration steps addresses some baseline issues, attention should be paid to the proper design of an OTF scan to provide frequent stabilizing observations.

3. Conclusions

In reducing STO-2 data a number of techniques were used to minimize stability issues common to heterodyne instruments. One approach to mitigate this issue was to make extensive use of the internal calibration sources. However, not all of STO-2 observations were taken in the OTF mode with frequent load calibrations. Those observations present an even greater calibration challenge.

The STO-2 mission was in preparation for the Gal/X-Gal Ultra long duration balloon Stratospheric Terahertz Observatory (GUSTO) which flies in 2021. GUSTO is a 100 day mission to survey the inner Milky Way at [CII], [NII] and [OI] transitions with heterodyne array receivers. To help make the GUSTO survey a success, we recommend standardizing observations utilizing frequent load calibrations as well as taking lessons from STO-2 about how to deal with instrument drifts in post processing.

References

- Allan, D. W. 1966, IEEE Proceedings, 54
- Kutner, M. L., & Ulich, B. L. 1981, ApJ, 250, 341
- Langer, W. D., Velusamy, T., Pineda, J. L., Willacy, K., & Goldsmith, P. F. 2014, A&A, 561, A122. 1312. 3320
- Mangum, J., Emerson, D., & Greisen, E. 2000, in Imaging at Radio through Submillimeter Wavelengths, edited by J. G. Mangum, & S. J. E. Radford, vol. 217 of Astronomical Society of the Pacific Conference Series, 179
- Ossenkopf, V. 2009, A&A, 495, 677. 0901. 2486
- Walker, C., Kulesa, C., Bernasconi, P., Eaton, H., Rolander, N., Groppi, C., Kloosterman, J., Cottam, T., Lesser, D., Martin, C., Stark, A., Neufeld, D., Lisse, C., Hollenbach, D., Kawamura, J., Goldsmith, P., Langer, W., Yorke, H., Sterne, J., Skalare, A., Mehdi, I., Weinreb, S., Kooi, J., Stutzki, J., Graf, U., Brasse, M., Honingh, C., Simon, R., Akyilmaz, M., Puetz, P., & Wolfire, M. 2010, in Ground-based and Airborne Telescopes III, vol. 7733 of Proceedings of the SPIE, 77330N

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

VO Service in Japan: Registry Service Based on Apache Solr and SIA v2 Service for Japanese Facilities

Yuji Shirasaki,¹ Christopher Zapart,¹ Masatoshi Ohishi,¹ and
Yoshihiko Mizumoto¹

¹*National Astronomical Observatory of Japan, Mitaka, Tokyo, Japan;*
yuji.shirasaki@nao.ac.jp

Abstract. More than 20 thousand VO services are registered in the VO registry. Keyword Search is the most popular way to find a resource. There can be a lot of ways to implement this capability and the performance depends on how to index the document describing the resource metadata. We upgraded the registry service behind the Japanese Virtual Observatory (JVO) portal to be based on Apache Solr – which is an open source search platform that uses the Lucene Java search library for full-text indexing. Under the Solr environment documents are indexed after the process that removes stop words and stems the word to a root word. Solr also can handle single- or multi-token synonyms and abbreviation. The new JVO registry service has increased the probability for a given keyword to hit a desired resource metadata. The data from Nobeyama Legacy project were released from the JVO portal on 1st June in 2018 and they are now also distributed through the most recent VO standard interface called Simple Image Access version 2 (SIA-v2). The data of ALMA and Subaru telescope are also accessible through SIA-v2 access.

1. Introduction

As the number of Virtual Observatory (VO) services is getting larger it becomes harder to find a resource that a user want to access. Although the standard interface for searching the resource metadata (RM) is defined in the International Virtual Observatory Alliance (IVOA) it is based on a kind of “strict matching criteria” which sometimes makes it difficult for a user to find the criteria that hit the desired resource metadata. What we want to have is a metadata service which supports a kind of “fuzzy search” method like a search engine such as Google. Another useful functionality is a “keyword suggestion” mechanism, by which users are directed to enter appropriate keywords. It is much simpler and more reliable for users to select from a list of predefined keywords than to type it by themselves. We constructed a VO RM service on top of the Japanese Virtual Observatory (JVO) portal ¹ (Shirasaki et al. 2017), which supports those features by defining the list of keywords extracted from the RMs and by adapting the Apache Solr document indexing tools.

We also operate VO services for distributing the data taken by Japanese facilities, Subaru Telescope, Nobeyama Radio Telescope, AKARI Satellite, and also the interna-

¹<http://jvo.nao.ac.jp/portal>

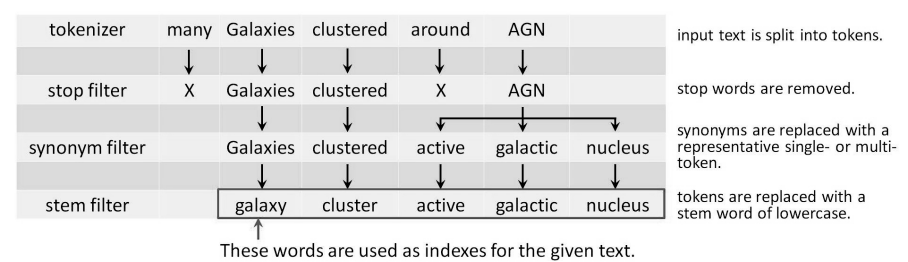


Figure 1. An example of how the text “many Galaxies clustered around AGN” is indexed.

tional facility ALMA. The most recent VO standard interface, Simple Image Access version 2 (SIA-v2), was adapted to our VO services by utilizing the JVO VO service toolkit (Shirasaki et al. 2012), which is described in the last part of this paper.

2. Solar based Registry Service

As described in the previous section, we defined a list of keywords and key phrases which are used as a classifier of the VO resources, that is category. Unfortunately some (or many) of the VO RMs are poorly fulfilled for the content metadata, we need to extract the keywords mostly from the description part of the document. The description is written in a natural language so lexical analysis needs to be applied to extract the keywords.

We used a Python implementation ² of the Rapid Automatic Keyword Extraction (RAKE) algorithm for the lexical analysis (Rose et al. 2010). Among a lot of keywords extracted from the RM, we selected ~800 primary keywords (categories) and ~1,000 secondary keywords which narrow down the contents of the primary category. This selection needed to be made by a human intervention, and this is a real bottleneck in creating and updating the category list.

We selected Apache Solr for the database of RMs. Apache Solr is an open source search platform and provides the functionality to develop the fuzzy search system. The text analyzer used in the Solr database system is customizable by selecting and combining many types of tokenizer and filters. Figure 1 shows an example of how a given text is indexed. As shown in the figure, it provides functionality to treat synonyms and stem the words which increase the probability to hit a desired RM for a given keyword. The caveat is one needs to create a synonym mapper file in advance to use the synonym filter.

Figure 2 shows the architecture of the JVO resource search system. The Solr-based RM database (registry) is located behind the JVO portal (Shirasaki et al. 2017), and the JVO portal provides a GUI for searching resources. The registry harvests RMs from the publishing registries distributed all over the world once per day.

²<https://github.com/aneesha/RAKE>

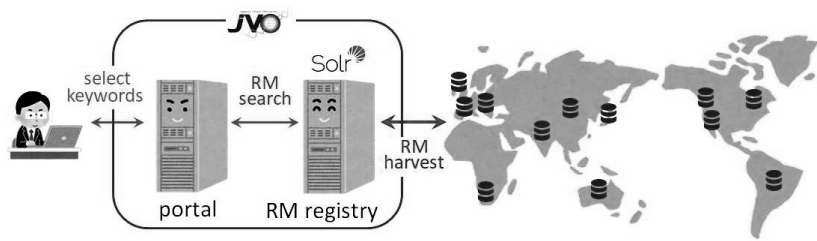


Figure 2. Architecture of the JVO resource search system.

Using the list of keywords extracted from the RMs and Solr-based RM registry we developed JVO index service. Figure 3 shows the GUI of the JVO index service. The keywords are displayed in the GUI in an alphabetic order and linked to the page listing the resources corresponding to (i.e. searched with) the keyword or key phrase. By looking over the list of keywords the user can easily learn what resource is available in the VO world which may help to bring out a new idea for his/her research.

3. VO Services of Japanese Facilities

Major data resources distributed through the JVO system are summarized in table 1 together with the VO standards adapted to the resource. All the services are accessible with TAP protocol. The most recent standards of SIA-v2 and ObsCore 1.1 are implemented in the ALMA, Suprime-Cam, and Nobeyama data services. JVO VO service toolkit (Shirasaki et al. 2012) was used to build these services.

Table 1. Major resources distributed with VO standard through the JVO system.

Title	SIA v1	SIA v2	SSA v1.1	TAP	Obs Core
ALMA VO Service		○		○	1.1
Subaru Suprime-Cam data service	○	○		○	1.1
Subaru MOIRCS data service	○			○	
Subaru HDS Spectrum data service			○	○	
Nobeyama Radio Telescope FITS archive		○		○	1.1
AKARI Far-infrared All-Sky Survey Maps	○			○	
AKARI Point Source Catalog Public Release 1				○	

JVO portal provides a dedicated search interface for the data of ALMA, Subaru Telescope, Nobeyama Radio Telescope, and Gaia Satellite. A web-based quick-look service, FITS WebQL (Eguchi et al. 2014; Zapart et al. 2019), is available to support the quick-look of the image and spectrum data. Also the desktop application Vissage (Kawasaki et al. 2017, 2019) is available for download ³. Vissage provides various dis-

³<http://jvo.nao.ac.jp/download/Vissage/>

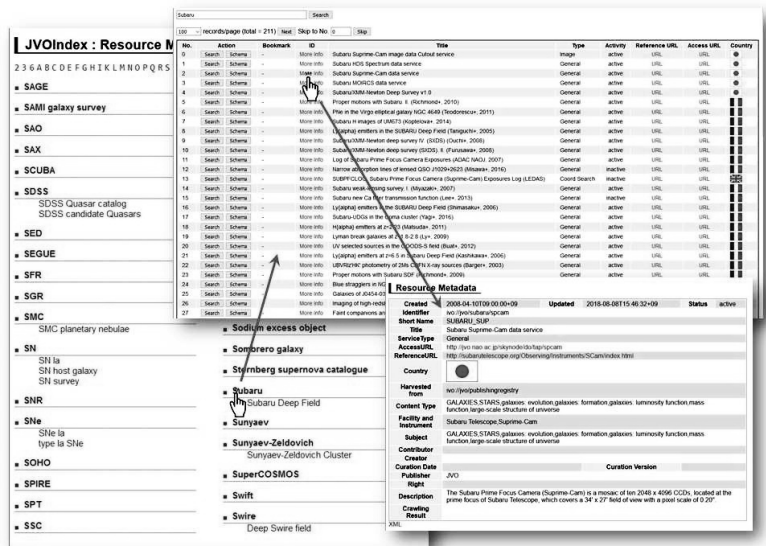


Figure 3. JVO Index service.

play modes, which is not provided on the FITS WebQL, such as P-V diagram, moment map, polarization map, and so on.

For the Gaia data search service we developed parallel database system to conduct an all-sky query in a reasonable time scale (a few tens minutes) for 1.7 billions records. We are now adapting this system to the Hyper Suprime-Cam Subaru Strategic Program (HSC-SSP) dataset⁴. The data of HSC-SSP (both image and catalog) will be distributed through the VO standard interface (SIA-v2 and TAP) hopefully before April 2019. We are also collaborating with JAXA to distribute the dataset obtained JAXA’s scientific satellites through the VO interface. We will begin with the dataset of Hitomi satellites.

References

Eguchi, S., et al. 2014, in ADASS XXIII, vol. 485 of ASP Conference Series, 7
Kawasaki, W., et al. 2017, in ADASS XXV, vol. 512 of ASP Conference Series, 617
— 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 37
Rose, S., Enger, D., Cramer, N., & Cowley, W. 2010, in Text Mining : Applications and Theory, edited by M. W. Berry, & J. Kogan (New York: John Wiley and Sons Ltd.), 3
Shirasaki, Y., et al. 2012, in ADASS XXI, vol. 461 of ASP Conference Series, 451
— 2017, in ADASS XXV, vol. 512 of ASP Conference Series, 585
Zapart, C., et al. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 13

⁴<https://hsc.mtk.nao.ac.jp/ssp/>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Running the Fermi Science Tools on Windows

Thomas E. Stephens

GSFC/Innovim, Greenbelt, MD, USA; thomas.stephens@nasa.gov

Abstract. The Fermi Science Tools, publicly released software for performing analysis on science data from NASA's Fermi Gamma-ray Space Telescope, have been available and supported on Linux and Mac OS since launch. Running the tools on a Windows based host has always required a virtual machine running Linux. New technologies, such as Docker and the Windows Subsystem for Linux, have made it possible to use these tools in a more native like environment.

In this paper we look at three different ways to run the Fermi Science Tools on Windows: via a VM, a Docker container, and using the Windows Subsystem for Linux. We present the steps necessary to install the tools, any issues or problems that exist, and benchmark the various installations. While not yet officially supported by the Fermi Science Support Center, these Windows installations are checked by staff when new releases are made.

1. Introduction

The Fermi Science Tools, now distributed as the FermiTools (see poster P4.3) are developed collaboratively between the Fermi instrument teams and the Fermi Science Support Center (FSSC). Development of these tools began before launch and the tools have been available since the very first public data release in 2009.

For a time, the Fermi Large Area Telescope (LAT) collaboration maintained a natively compiled version of the tools for Windows. This version regularly had issues and was never publicly released or supported. It was discontinued in 2014.

While it has always been possible to run the tools on Windows using a VM with a Linux OS, this is a somewhat "heavy" solution. With the recent development of both Docker and the Windows Subsystem for Linux, more lightweight options are possible that allow Windows users to use the supported Linux version of the FermiTools in a more native fashion.

In this paper we discuss the installation and use of the FermiTools in three different scenarios, look at performance benchmarks, and discuss any associated issues or caveats.

2. Installation

2.1. Virtual Machine

Using a VM to run the tools has been possible since launch. For this study, we used two different VMs both running under Oracle's Virtual Box system. One ran Ubuntu 18.04 OS and the other ran Scientific Linux 7.5.

Of the three methods, a virtual machine installation is the most straightforward but also the most resource intensive, requiring the most disk space to hold the virtual machine image. Installation consists of five steps:

1. Create VM and install the guest OS
2. Install a C/C++ compiler
3. Install Anaconda/Miniconda
4. Install the FermiTools
5. Within the FermiTools Conda environment, install pyds9 via pip.

As an optional, although desired, step, one can set up a shared directory between the guest and host operating systems to store the data being analyzed. While not necessary, if Windows is your primary OS where your other tools reside, this may be a desirable configuration.

2.2. Docker

Docker provides a lighter-weight alternative to installing a full VM for running software that has been bundled into an appropriate Docker container. Since we are running Linux software on a Windows kernel, this is not as light-weight as it could be and is essentially a transparent way to run VM without having to set it up manually.

Starting with the 2018 Fermi Summer School, the FSSC began providing a pre-built Docker container with the current FermiTools version. Based on CentOS 6, this container is available in the fssc/fermibottle repository on Docker Hub.

For this installation method, no setup within the container is needed. Simply run the container and attach to it and the tools are ready to go. The one additional installation step is to install an Xwindows system on the Windows host. This is only needed if you intend to use the graphics capabilities of the tools (e.g. plotting or GUI interfaces).

The one caveat with this method is that Docker for Windows, the default Docker system, is only available for the Professional (and Enterprise) version of Windows. If you are running Window Home, you have to install the older Docker Toolkit.

2.3. Windows Subsystem for Linux

While potentially the most seamless and "native" feeling of the installation options, using the Windows Subsystem for Linux (WSL) requires the most initial setup. It has an advantage over Docker, however, in that it is available on any Windows OS, not just the Professional versions. To install and use the FermiTools in a WSL environment:

1. Activate the WSL feature in your OS
2. Install a Linux distribution from the Windows store (we tested Ubuntu 18.04)
3. Install an XWindows server if desired (as for Docker)
4. Within the Linux environment perform the same installation steps as for the Linux VM (steps 2-5)

3. Benchmarks

All of the benchmarking tests were run on the same system, a Dell 7250 Workstation laptop with an Intel Xeon E3-1505M v6 CPU (4 Hyper-threaded cores, 3.0 GHz) and 32 GB of RAM. For testing, we used test scripts that run the unit tests for the FermiTools and that implement the binned and unbinned likelihood analysis from the Fermi Analysis Threads (<https://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/>). These tests ensure that all the functionality is working and exercise the tools in standard analysis scenarios.

Figures 1a, 1b, & 1c show the timing results for the three different tests on the four different systems. Each test was run 10 times on each system.

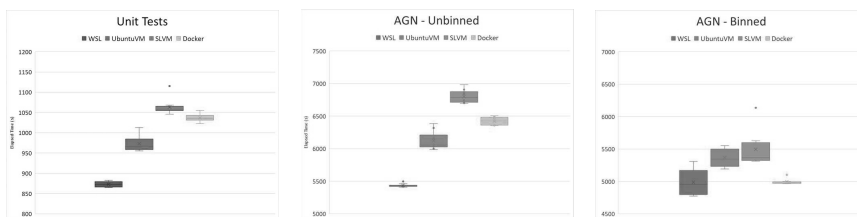


Figure 1. Box and whisker charts showing the spread of execution times for the tests on the various systems

While the tests were typically run when the system was not being used for much else, and only consumed resources on one or two of the system's 8 virtual cores, some of the scatter seen is due to load from other processes on the machine. However, since this will often be the case when analyses are run with the tools, this is not really an issue. The narrower box plots for some tests are due to those tests running overnight when the system had minimal load.

We notice a few things right away. First the general trend is the same for all three tests. In fact, with the exception of the AGN analysis with binned likelihood, the relative ranking of the four systems is the same across the tests.

The Scientific Linux VM is the slowest of all the systems. While this VM was not a fresh install of the OS but one used regularly for work, the fact that the Docker container built on the same OS was the second slowest in two of the three tests would seem to indicate an issue with the OS itself.

The Windows Subsystem for Linux installation running Ubuntu 18.04 was by far and away the fastest of the installations across all the tests, being 10-20% faster than the SL VM depending on the test. Since the Ubuntu VM was faster than the SL VM, some of that speed up is due to the OS but in every test, the WSL Ubuntu installation out-performed the VM installation by 7-12%. Similarly, the Docker installation always outperformed the VM installation of Scientific Linux although not by quite the same margin (only 3-9%).

4. Issues, Pros, & Cons

All three installation methods worked just fine for the science analysis tests. Each method also has its advantages and disadvantages in relation to setup and operation.

4.1. Virtual Machines

The main advantage of the VM install is its familiarity and direct correspondence to a standard install. Once the VM is up and running, all the tools, instructions, and help for the FermiTools apply since you are effectively working in the native environment.

The main downside is resource usage. Of the three options, this installation is the "heaviest". It requires the most disk space for the virtual disks and there is a larger overhead for the virtual machine manager. Anecdotally (I didn't make hard measurements but was constantly monitoring CPU usage), the tests running in the VM used 3-5% more CPU than the same test running in Docker or WSL.

4.2. Docker

The first time running the Docker environment tests, the Binned AGN test failed due to lack of memory. By default, Docker gives the VMs it creates a 2GB memory limit. This was not enough to run the test. In order to analyze larger datasets with this installation method, you need to increase the memory limit in Docker. The benchmarks presented were run with an 8GB limit and had no issues. Further investigation should be done to quantify exactly how much memory is needed for different analysis scenarios.

The main downside to this method is probably unfamiliarity with Docker and running and connecting to Docker images. Once that hurdle is overcome, this is one of the easiest methods as the tools are set up in the Docker image and can be used right away.

4.3. Windows Subsystem for Linux

One downside of using WSL for the FermiTools is that it has the most complicated setup. However, once set up, it is the easiest to use; all you have to do is launch the Linux app to get your shell, start the Conda environment and you're off to the races. It also provides native integration with the host file system.

One issue discovered during testing is a problem with how WSL handles the local time zone. While the other methods use standard Linux time zone data, WSL creates an internal time zone based on the user's system clock. This manifested by some unit test errors that were off by an hour due to daylight savings time. This, however, only affects time values in the FITS headers and does not affect the science analysis. This issue has since been handled in the tools.

5. Discussion

The use of Conda for distribution of the FermiTools greatly eases the problem of using them for scientific analysis on Windows. Because the binaries are precompiled and distributed with all of their dependencies, users do not have to worry about configuration and compilation.

This new distribution method, combined with technology advances that provide more and easier ways to run and use Linux software on Windows, now makes that OS a viable platform for Fermi scientific analysis with only a bit of extra set up.

Of the methods tested, Microsoft's Windows Subsystem for Linux, while requiring the most effort to set up and install, provides the fastest processing while the Docker container requires the least user setup after getting the hosting environment configured.

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Binospec@MMT: A Database Driven Model of Operations, from Planning of Observations to Data Reduction and Archiving

Igor Chilingarian,^{1,2} Sean Moran,¹ Martin Paegert,¹ Daniel Fabricant,¹ Jan Kinsky,¹ and Warren Brown¹

¹*Smithsonian Astrophysical Observatory, 60 Garden St., Cambridge MA 02138, USA; igor.chilingarian@cfa.harvard.edu*

²*Sternberg Astronomical Institute, M.V. Lomonosov Moscow State University, 13 Universitetsky prospect, Moscow 119234, Russia*

Abstract. Binospec is a new optical multi-object spectrograph operated at the 6.5-m MMT at Mt.Hopkins, Arizona since Nov/2017. Here we describe the Binospec software system driven by the PostgreSQL relational database that covers all stages of the instrument operations from the slit mask design to the data reduction pipeline and archiving and distribution of raw and reduced datasets. We developed a web application to design slit masks, which submits configurations directly into the database and sends them to the mask cutting facility. When masks are being installed in the spectrograph, the process is logged in the same database that also connected to the MMT telescope scheduler and Binospec instrument control software. We record metadata for all exposures collected with Binospec in real time, then automatically reduce the data with a dedicated pipeline, then ingest the metadata of reduced datasets and finally distribute data products to the PIs. Using currently available technologies and very limited manpower we built a complete database driven software system similar to those deployed by major space missions like Chandra, XMM, and HST.

1. Motivation for Database Driven Operations and the Binospec Database

Efficient operations of modern instruments at mid-size and large telescopes represent a challenge especially when operational budgets are tight and manpower is limited. It becomes crucial to connect all stages of operations, from the preparation of an observing run and executing observations to the data reduction and distribution, into a single workflow. Making every component of a software system talk to the same database is one of possible solutions.

Binospec is a new optical dual channel high-throughput multi-object spectrograph operated at the f/5 Cassegrain focus of the 6.5-m converted MMT telescope at Mt.Hopkins, Arizona. The instrument was commissioned in Nov-Dec/2017 and started routine operations in early 2018. Yet at the construction stage we decided to exploit a database driven model of operations, which had been successfully used by several space missions and the Atacama Large Millimeter Array (ALMA).

All stages of Binospec operations are centered (Fig. 1) around the Binospec database (BinospecDB) implemented using *PostgreSQL* relational database management system. We use *pgSphere* (Chilingarian et al. 2004) to handle astronomical coordinates.

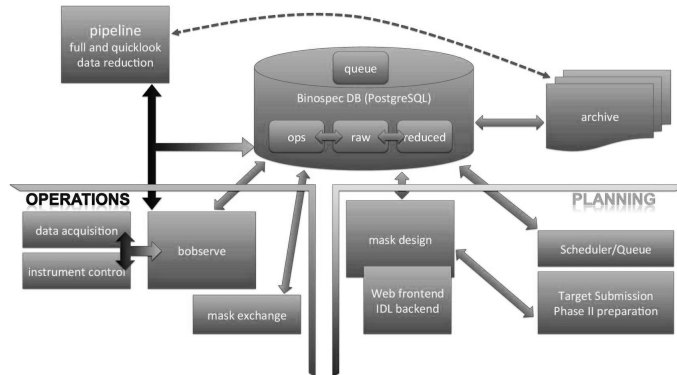


Figure 1. Binospec operations are centered around and driven by an SQL database.

BinospecDB has several loosely connected groups of tightly linked tables designed for certain aspects of operations and data management implemented as SQL schemata:

- *binospec_config* – information on the state of Binospec and its control software
- *binospec_ops* – a schema updated in real-time that contains all the information required for scientific operations of the instrument at the telescope, such as current configuration of slit masks, including information on science targets, guide stars, and wavefront sensor stars for each. It also contains complete metadata of each image saved to disk along with optional queue observer notes
- *binospec_archive_raw* and *binospec_archive_reduced* – metadata of raw and reduced archived datasets that is updated during observations with the information from the *binospec_ops* schema and once the Binospec pipeline has been executed
- *scheduler* – a schema maintained by the MMT Queue Scheduler software team, which contains all targets submitted to the queue, including information on exposure time, instrument setup, any special requests

2. Planning of Observations

The observation planning (also known as *phase-2* submission) is started once the telescope time has been allocated to a program by the time allocation committee of one of the MMT consortium partners (Smithsonian Astrophysical Observatory or the University of Arizona). A web-based platform developed at the MMT Observatory allows program PIs to submit targets into BinospecDB for one of the following three types of observations: (i) imaging; (ii) long-slit spectroscopy; (iii) multi-object slit mask spectroscopy (MOS). The two former cases require standard parameters like coordinates, position angle of the field, filter, disperser, slit type, length of individual exposures and the number of exposures.

The MOS mode requires an extra step that is a slit mask preparation. The mask design software is implemented as the interactive web application BinoMask¹ that uses

¹<https://scheduler.mmt.az.arizona.edu/BinoMask/index.php>

the *Bootstrap v3* JavaScript library and *JS9* as a FITS display tool. It accepts lists of targets with different priorities, observational constraints (approximate date and time) as .csv files and then invokes a server-side IDL code to place the slits. Then the optimally placed slits are returned as *json* and displayed in the browser. Once the slit configuration has been approved by a user, it is submitted directly to the *binospec_ops* schema. Every individual target as well as the exact geometry of a corresponding slit in a mask is stored in the database. Then this information is used to generate .*dx* files used at the laser slit cutter. At the end, a reference to a newly designed slit mask is added to the *binospec_ops* schema.

3. Operations

Before the start of an observing run, the *scheduler* software developed by the MMT Observatory is run. It collects all submitted targets from BinospecDB and prepares a preliminary schedule for the entire duration of the run. Each target is checked against observing constraints set by the PI such as the Moon presence in the sky and phase, the airmass, and the ability to schedule an observation of certain duration. At the end it is split into observing blocks (OBs), relatively short sequences (typically under 2h) with all necessary calibrations, which can later be reduced and analyzed as independent observations even if conditions/weather prevented the execution of the remaining observations for a given target.

Then, for every night of an observing run, a list of OBs is generated based on the scheduler information. The masks needed for the OBs of the upcoming night are loaded into the instrument during the day and this process is logged in BinospecDB. Then the *MMT dispatcher* software generates a short-term queue schedule for one night based on the scheduling information for the entire observing run, the percentage of completed OB for every program, visibility and Moon constraints, and the list of installed masks. All MMT queue software is web based.

Observations are performed using the *bobserve* instrument control and data acquisition software. An observer sends an OB from the queue prepared by *MMT dispatcher* to *bobserve*, then an observing sequence with science and calibration frames is automatically generated and executed when the telescope is in position and the guiding and wavefront sensor corrections are on. Every frame collected from the instrument is automatically logged in BinospecDB. Spectral frames have two extra FITS binary table extensions, which contain slit mask descriptions.

Once an OB has been completed, *bobserve* generates a script for the Binospec pipeline that can be used to perform a quicklook or full data reduction at the telescope.

4. Data Reduction Pipeline

The Binospec data reduction pipeline is an open-source IDL software package distributed under the GPLv3 license². The pipeline can reduce spectroscopic data collected in any mode supported by Binospec.

The Binospec pipeline is based on the MMT and Magellan Infrared Spectrograph (MMIRS) data reduction pipeline (Chilingarian et al. 2015), most of which are run

²https://bitbucket.org/chil_sai/binospec

twice because the data from each of the two Binospec channels is handled independently. The pipeline includes the following steps:

- *Primary data reduction.* Each of the two Binospec cameras is equipped with a CCD read through 4 amplifiers; hence each raw image is a FITS file with 8 image extensions. The pipeline mosaics it into two images (one per channel) taking into account the amplifier gain differences and correcting for the non-linearity of the signal using lab calibration data. Cosmic ray hits are also corrected at this step.
- *Tracing, Extraction of 2D Slits, and Flat Fielding.* Using internal spectral field frames and the slit mask description attached to raw images as FITS binary tables, the pipeline identifies regions on a CCD that contain a spectral trace of every slit. Then they are extracted from science and calibration images and stored in a multi-extension FITS file (one slit trace per extension). Then, a flat field frame is used to correct pixel-to-pixel variations and illumination in the data.
- *Wavelength calibration.* To predict positions of arc lines we first use an analytic 3D grating equation, which we calibrated to about 1 pix. Then we identify detected arc lines against NIST line lists in every slit and build polynomial wavelength solutions. Finally, we fit all line positions in all slits simultaneously using a 3-dimensional polynomial fit of a scattered dataset in the form: $\lambda = D(x_{\text{pix}}, x_{\text{mask}}, y_{\text{mask}})$. This fit is accurate to about 1/40 pix.
- *Sky subtraction.* We use the Kelson (2003) approach to subtract the night sky emission for long slit data and its non-local form proposed by us for the MMIRS pipeline, that yields the Poisson-limited quality of sky subtraction in most cases.
- *Linearization, flux calibration.* The spectra from every slit are rectified and linearized in wavelength using the polynomial fit of the wavelength solution. Then we correct them for the atmospheric extinction and perform flux calibration using sensitivity curves derived from observations of spectrophotometric standards.
- *Extraction of one-dimensional spectra.* At the final stage, we extract 1D spectra from flux calibrated 2D slits. We use either a box kernel or perform the optimal extraction with a profile determined empirically from the data for brighter objects or assuming it to be Gaussian for fainter targets.

5. Archiving and Data Distribution

Because the metadata of raw and reduced datasets is stored in BinospecDB, it is trivial to hook the reduced datasets into the system. Currently, we distribute data products using the MMT scheduler web-site and inform the PIs by e-mail when the datasets have been linked to the corresponding proposed targets on their personal page.

References

- Chilingarian, I., Bartunov, O., Richter, J., & Sigaev, T. 2004, in *Astronomical Data Analysis Software and Systems (ADASS) XIII*, edited by F. Ochsenbein, M. G. Allen, & D. Egret, vol. 314 of *Astronomical Society of the Pacific Conference Series*, 225
- Chilingarian, I., Beletsky, Y., Moran, S., Brown, W., McLeod, B., & Fabricant, D. 2015, *PASP*, 127, 406. 1503.07504
- Kelson, D. D. 2003, *PASP*, 115, 688. astro-ph/0303507

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

QAC: Quick Array Combinations with CASA

Peter Teuben

Astronomy Department, University of Maryland, College Park, MD, USA
teuben@astro.umd.edu

Abstract. QAC is a simple python layer in CASA, developed to aid writing scripts for radio astronomy array (single dish and interferometric) combinations and simulations. Although initially developed for TP2VIS, running simulations and comparing with other array combination methods, this package turned out to be useful for array design studies as well. Both ALMA and ngVLA simulations are supported, but extending to more generic array are planned. QAC may be less useful for real data, where more CASA flexibility might be needed.

1. Introduction

CASA (McMullin et al. (2007), and in this volume Emonts (2019)) is a general purpose python interface to radio astronomy software. It handles interferometric as well as single dish data, all the way from ingestion, calibration, and mapping to analysis. Most ALMA and VLA data are now routinely processed with CASA using a custom built pipeline. CASA uses object oriented “tools”, as well as the more classic python functions, called “tasks” in CASA. One can write very complex procedures this way, and in fact, the ALMA/VLA pipeline is an example of such an interface. The QAC interfaces discussed in this paper was also designed with a specific goal of testing the combination of single dish and interferometric data. They are also a convenient method to study array design (Turner et al. 2019).

The development of QAC started in 2017 with the TP2VIS project (Koda et al. 2019), to provide a more easily scriptable interface, orchestrate simulations and provide a reproducible baseline using regressions. It can be obtained from <https://github.com/teuben/QAC>. We first summarize the different methods how CASA can be extended by using your own custom built python code, then how QAC is installed and used, with some examples.

Several other efforts have been going on wrapping CASA tasks and tools in a more convenient environments, e.g. *casanova*¹, *ADMIT*², *RICA*³, and of course the calibration and imaging pipelines for ALMA and VLA⁴.

¹<https://github.com/kaspervd/casanova>

²<http://admit.astro.umd.edu/>

³<https://gitlab.com/miles-lucas/RICA>

⁴Included with CASA via https://casa.nrao.edu/casa_obtaining.shtml

2. Running python code in CASA

CASA interacts with the user in a modified interactive python (ipython) session. For most users adding C++ code is a complex operation, but installing a new python interface is usually fairly straightforward, and nowadays most users are familiar with this. Several methods (and hybrids between these) exist for CASA⁵:

1. `buildmytasks`

This is the native CASA method of installing a conforming CASA task. The CASA Cookbook describes a procedure to install new CASA tasks, but at the same time warns this method may get deprecated. Nonetheless, this “buildmytasks” has been used by other teams, most notably by the Nordic ARC node⁶. This is typically run once inside the directory where your `foo.py`, `foo.xml`, and other material is present, after which the code, and documentation, gets installed at the right place inside the CASA tree. Users can then run this task with the CASA command

```
foo(1, 'b', [1,2,3])
```

but since this is now a true CASA task, commands such as `inp` will also work.

2. `import foo`

The traditional “pythonic” way a user includes software would be the `import` command. This is fine for stable software, and can be installed with python’s `setuptools` in CASA. In the future CASA6 one should be able to use `virtualenv` to test out software like this without the need to write into CASA’s personal space. One can also consider the use of using `$PYTHONPATH` to point to the directory where `foo.py` is present, but this method can easily conflict with other installation methods (in fact, is strongly discouraged in a CASA environment).

```
foo.bar(1, 'b', [1,2,3])
```

In this, and all following examples below, `foo()` is just a python function, not a CASA task.

3. `execfile('foo.py')`

This will execute the named code, after which variables and functions (the API) is immediately available for use. (note this will not work in python3 anymore). This is the method used in QAC. Notice that this way no command line parameters can be passed into the code, see below for another approach to this.

```
foo_bar(1, 'b', [1,2,3])
```

⁵some of these methods are expected to become more common in CASA6

⁶<https://www.oso.nordic-alma.se/software-tools.php>

4. `run foo.py p1 p2 p3`

Since CASA is essentially an ipython shell, the `run` command can be used to execute a script, including conveniently parsing “command line arguments”. This will then need a parser to bind `p1=sys.argv[1]`, `p2=sys.argv[2]`, etc. Note this is an ipython interface, not python, though it’s similar to running in the unix shell `python foo.py p1 p2 p3`. The aforementioned RICA package uses this method.

```
run foo.py 1 'b' [1,2,3]
```

5. `%run -m foo`

Runs the `foo` module (from `sys.path`). In the current CASA manipulating python’s `sys.path` is not recommended, the arguments similar to those of not using `$PYTHONPATH`

3. Using QAC

In order to use QAC, CASA needs to be installed first, although there is also a self-contained option within QAC to install CASA. Since CASA’s startup can be controlled by `~/casa/init.py` we choose this file to `execfile()` the correct startup script, aptly named `casa.init.py` in the QAC distribution:

```
execfile(os.environ['HOME'] + '/.casa/QAC/casa.init.py')
```

this file also contains a few other common examples how packages are loaded by CASA as discussed in the previous section.

Here is an example of the `simplenoise` procedure where one desires to add a given noise to a simulated dataset. QAC calls the VLA simulator `qac_vla()` twice, in the end generating a Measurement Set (MS) with the correct 2 mJy/beam noise. In the example below for a given model the directory `pdir` will contain all the results:

```
rms = 0.002                                # requested 2 mJy/beam RMS noise
ms1 = qac_vla(pdir,model, noise=-rms)       # noise<0 triggers computing the rms
sn0 = qac_noise(noise,pdir+'/noise', ms1)   # get scaling factor from rms in ms1
ms2 = qac_vla(pdir,model, noise=sn0)        # MS with correct "rms" in Jy/beam
qac_clean1(pdir+'/clean1',ms2)              # image and clean the MS
```

In `ms1` a noise level is computed for a fixed 1 Jy noise per visibility on a zero sky model. The noise in the resulting dirty map, computed in `qac_noise()`, is then the scaling factor (`sn0`) that needs to be applied to get the correct requested noise level in `ms2`, which can then be used for mapping.

Finally, a convenient way to run QAC scripts could also be from the Unix command line (or via Makefile), e.g.

```
casa --nogui -c vla1.py pixel_m=0.5 niter='[0,500,1500]' dish=45 pdir="exp12"
```


4. Timing, Benchmarks, and Regression

QAC also adds support to time-stamp your code, run benchmarks, add regression etc. Since QAC deals almost exclusively with image type data, the regression test is invoked automatically with the image statistics report if a regression string is given:

```
reg_54 = "0.00383 0.021439 -0.048513 0.41929 383.60327"
qac_stats(test+'/clean/tpint.image')
qac_stats(test+'/clean/tpint_4.tweak.image', reg_54, eps=0.001)
```

where in the first instance only the statistics are reported, the second instance will also flag any deviations from that expected series of numbers. The numbers represent the mean, std, min, max and total flux of the image. One of the options is to regress these values within a relative accuracy of `eps`.

5. Future

CASA is a development project, the next release (V6) will have a major overhaul how python and the C++ libraries are integrated, and this will likely have some effect how QAC is installed, although less on its API. Ideally we like to switch to the `import` or `run` method once the CASA imports are standardized.

Although QAC is very convenient to write more compact scripts, a drawback we quickly ran into were CASA bugs. Any complex situation would need to be translated in pure-CASA examples to the CASA developers. Finally, QAC is great for simulations, but does not always expose all the rich parameters that full CASA tasks have, which could be a problem if you wanted to use QAC for real telescope data.

Acknowledgments. Jordan Turner and Sara Negussie have been patient contributors and users. Part of QAC was developed under the ALMA development study “TP2VIS” and the “ngVLA” array combination study (ngVLA memo 54).

References

- Emonts, B. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 265
- Koda, J., Teuben, P., Sawada, T., Plunkett, A., & Fomalont, E. 2019, Publications of the Astronomical Society of the Pacific, in press.
- McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in Astronomical Data Analysis Software and Systems XVI, edited by R. A. Shaw, F. Hill, & D. J. Bell, vol. 376 of Astronomical Society of the Pacific Conference Series, 127
- Turner, J., Teuben, P., & Dale, D. 2019, Short Spacing Issues for Mapping Extended Emission: Milky Way Case Study, Tech. Rep. 54. URL https://library.nrao.edu/public/memos/ngvla/NGVLA_54.pdf

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Super-resolution Imaging of the Protoplanetary Disk HD 142527 using Sparse Modeling

Masayuki Yamaguchi,^{1,2} Kazunori Akiyama,^{2,3,4,5} Akimasa Kataoka,^{2,7}
Takashi Tsukagoshi,² Takayuki Muto,⁹ Shiro Ikeda,^{6,7} Misato Fukagawa,²
Mareki Honma,^{2,7} and Ryohei Kawabe^{1,2,7}

¹*Department of Astronomy, Graduate School of Science, The University of
Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan;*
masayuki.yamaguchi@nao.ac.jp

²*National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo
181-8588, Japan*

³*National Radio Astronomy Observatory, 520 Edgemont Rd, Charlottesville,
VA 22903, USA*

⁴*Massachusetts Institute of Technology, Haystack Observatory, 99 Millstone
Rd, Westford, MA 01886, USA*

⁵*Black Hole Initiative, Harvard University, 20 Garden Street, Cambridge, MA
02138, USA*

⁶*The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo
190-8562, Japan*

⁷*Department of Statistical Science, School of Multidisciplinary Sciences,
Graduate University for Advanced Studies, 10-3 Midori-cho, Tachikawa, Tokyo
190-8562, Japan*

⁸*Department of Astronomical Science, School of Physical Sciences, Graduate
University for Advanced Studies, 2-21-1 Osawa, Mitaka, Tokyo 181-8588,
Japan*

⁹*Division of Liberal Arts, Kogakuin University, 1-24-2 Nishi-Shinjuku,
Shinjuku-ku, Tokyo 163-8677, Japan*

Abstract. High-resolution observations of protoplanetary disks with radio interferometers are crucial for understanding the planet formation process. Recent observations using Atacama Large Millimeter/submillimeter Array (ALMA) have revealed various small-scale structures in disks. In interferometric observations, the observed data are an incomplete set of Fourier components of the radio source image. The image reconstruction is therefore essential in obtaining the images in real space. The CLEAN technique has been widely used, but recently, a new technique using the sparse modeling approach is suggested. This technique directly solves a set of undetermined equations and has been shown to behave better than the CLEAN technique based on mock observations with VLBI (Very Long Baseline Interferometry). However, it has never been applied to ALMA-like connected interferometers nor real observational data. In this work, for the first time, the sparse modeling technique is applied to observational data sets taken by ALMA. We evaluate the performance of the technique by comparing

the resulting images with those derived by the CLEAN technique. We use two sets of ALMA archival data at Band 7 (~ 350 GHz) for the protoplanetary disk around HD 142527. One is taken in the intermediate-baseline array configuration, and the other is in the longer-baseline array configuration. The image resolutions reconstructed from these data sets are different by a factor of ~ 3 . We compare images reconstructed using sparse modeling and CLEAN. We find that the sparse modeling technique can successfully reconstruct the overall disk emission. The previously known disk structures appear on both images made by the sparse modeling and CLEAN at its nominal resolutions. Remarkably, the image reconstructed from intermediate-baseline data using the sparse modeling technique matches very well with that obtained from longer-baseline data using the CLEAN technique.

1. Introduction

Finer resolution with radio interferometers providing a more detailed picture of disk structure can constrain their evolution and establish the framework of this type. Recent observations using Atacama Large Millimeter/submillimeter Array (ALMA) have revealed various small-scale structures in disks. In interferometric observations, images of astronomical source $I(l, m)$ can be obtained by two-dimensional Fourier transform of observed data $V(u, v)$. However, in practical observation, there must be space between antennas, which causes unsampled holes in the (u, v) plane. Such an incomplete (u, v) coverage always causes “underdetermined problem” in the radio interferometer equation. Usually, this problem is solved by filling unsampled visibilities with zero. However, this process causes the resultant image to be “dirty image”.

2. Imaging Techniques for Radio Interferometer

2.1. Imaging with CLEAN

CLEAN (Högbom 1974) is a most common algorithm in radio astronomy and uses nonlinear techniques effectively interpolate sample of $V(u, v)$ into unsampled regions of the (u, v) plane. The algorithm assumes that the image consists of many point sources and is, therefore, to break down the intensity distribution in the dirty image into point source responses, and then replaces each one with the Gaussian with a half-amplitude width equal to that of the original synthesized beam. This procedure finally provides a CLEAN image, but there is a limitation of the nominal resolution (diffraction limit) defined by the maximum length of the baseline between two telescopes. In this work, multi-scale CLEAN algorithm (henceforth MS-CLEAN; Cornwell 2008; Rau & Cornwell 2011) is adopted as the image reconstruction. Data were calibrated by using the Common Astronomy Software Applications (CASA) package (CASA; McMullin et al. 2007).

2.2. Imaging with Sparse Modeling

For the imaging with sparse modeling, ℓ_1 +TSV regularization (Kuramochi et al. 2018) is adopted as the regularization function of the image reconstruction. The equation becomes a convex optimization, guaranteeing the global convergence to a unique solution regardless of initial conditions by choosing the meaningful sparse image from the infi-

nite number of possible solutions by introducing two regularization terms: the ℓ_1 norm and Total Squared Variation (TSV) of the brightness distribution (see Akiyama et al. 2017b; Kuramochi et al. 2018, for details), and can achieve an optimal resolution of $\sim 30\%$ of the diffraction limit (~ 3 time better angular resolution) by adjusting two positive variables Λ_ℓ and Λ_t in these two regularization terms. Λ_ℓ and Λ_t can effectively be determined by evaluating goodness-fitting with 10- fold cross validation (e.g., Honma et al. 2014; Akiyama et al. 2017a,b; Kuramochi et al. 2018)

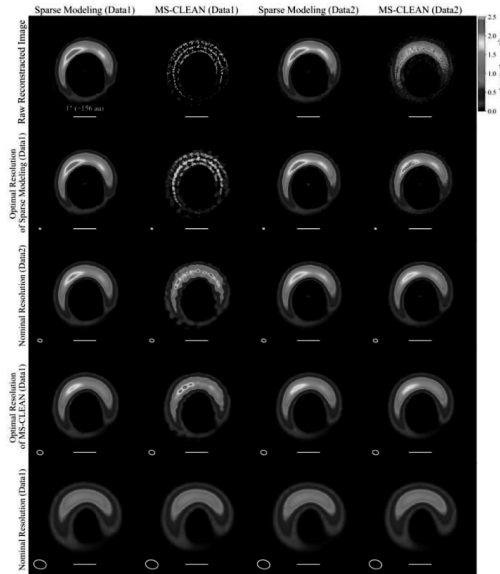


Figure 1. Images of HD 142527 from two datasets of ALMA observations at Data 1 and Data 2 reconstructed with sparse modeling and MS-CLEAN. The left two columns show images from Data 1 reconstructed with sparse modeling and MS-CLEAN, respectively, while the right two columns show images from Data 2 as well. Each row shows raw reconstructed images or those restored with an elliptical Gaussian beam, whose FWHM shape is shown in a yellow ellipse in each panel. (2nd/4th panels): Reconstructed images convolved with the optimal beam of Data 1 for sparse modeling and MS-CLEAN, respectively, determined by NRMSE analysis (Chael et al. 2016; Akiyama et al. 2017b; Kuramochi et al. 2018). (3rd/5th panels): Reconstructed images convolved with the nominal beams, which have the same solid angles same to the synthesized beam of Data 1 and 2, respectively, for the above Briggs weighting.

2.3. Analysis

We adopted two data sets of ALMA observations toward the protoplanetary disk HD 142527 at close frequencies in Band 7 using different array configurations. The maximum baseline lengths of these data sets differ by a factor of ~ 3 . Throughout this paper, we name the data set obtained with the more compact (i.e., intermediate-baseline) array configuration as *Data 1*, while that from the more extended (i.e., longer-baseline) one as *Data 2*. In below, we summarize observations for each data set. Data 1 was

obtained as part of the project 2015.100425.S, which are already reported in Kataoka et al. (2016). The corresponding observations were carried out on March 11, 2015, at 343 GHz (0.87 mm) and at full polarizations. The observing array consisted of thirty-eight 12 m antennas, providing the longest projected baseline length of 460 m. Data 2 were obtained as part of project 2012.1.00631.S on 2015 July 17 at 322 GHz (0.93 mm) and at dual polarizations. Observations made use of forty 12 m antennas with the longest projected baseline length of 1570 m. For the data sets, We compare images reconstructed by sparse modeling and CLEAN.

3. Results

3.1. Image Appearance at Different Angular Resolution

As shown in Figure 1, we find that the sparse modeling technique can successfully reconstruct the overall disk emission. The previously known disk structures appear on both images made by the sparse modeling and CLEAN at its nominal resolutions. Remarkably, the image reconstructed from Data 1 using the sparse modeling technique matches very well with that obtained from Data 2 using the CLEAN technique with the accuracy of $\sim 90\%$ on the image domain (see, the third panel of Figure 1).

4. Conclusion

We have shown that the sparse modeling technique is potentially useful in actual data analyses and may improve the spatial resolution by a factor of 3.

References

- Akiyama, K., Ikeda, S., Pleau, M., Fish, V. L., Tazaki, F., Kuramochi, K., Broderick, A. E., Dexter, J., Mościbrodzka, M., Gowanlock, M., Honma, M., & Doeleman, S. S. 2017a, *AJ*, 153, 159. 1702.00424
- Akiyama, K., Kuramochi, K., Ikeda, S., Fish, V. L., Tazaki, F., Honma, M., Doeleman, S. S., Broderick, A. E., Dexter, J., Mościbrodzka, M., Bouman, K. L., Chael, A. A., & Zaizen, M. 2017b, *ApJ*, 838, 1. 1702.07361
- Chael, A. A., Johnson, M. D., Narayan, R., Doeleman, S. S., Wardle, J. F. C., & Bouman, K. L. 2016, *ApJ*, 829, 11. 1605.06156
- Cornwell, T. J. 2008, *IEEE Journal of Selected Topics in Signal Processing*, 2, 793
- Högbom, J. A. 1974, *A&AS*, 15, 417
- Honma, M., Akiyama, K., Uemura, M., & Ikeda, S. 2014, *PASJ*, 66, 95
- Kataoka, A., Tsukagoshi, T., Momose, M., Nagai, H., Muto, T., Dullemond, C. P., Pohl, A., Fukagawa, M., Shibai, H., Hanawa, T., & Murakawa, K. 2016, *ApJ*, 831, L12. 1610.06318
- Kuramochi, K., Akiyama, K., Ikeda, S., Tazaki, F., Fish, V. L., Pu, H.-Y., Asada, K., & Honma, M. 2018, *ApJ*, 858, 56
- McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in *Astronomical Data Analysis Software and Systems XVI*, edited by R. A. Shaw, F. Hill, & D. J. Bell, vol. 376, 127
- Rau, U., & Cornwell, T. J. 2011, *A&A*, 532, A71. 1106.2745

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

MIRISim: The JWST-MIRI Simulator

Vincent C. Geers,¹ Pamela D. Klaassen,¹ Steven Beard,¹ and
 MIRI European Consortium²

¹*UK Astronomy Technology Centre, Blackford Hill, Edinburgh, EH9 3HJ,
 United Kingdom; vincent.geers@stfc.ac.uk*

²*jwst-miri.roe.ac.uk*

Abstract. MIRISim is the simulator package for the Mid-Infrared Instrument (MIRI) on the James Webb Space Telescope (JWST). MIRISim simulates photon propagation through MIRI and delivers detector images consistent with the expected on-orbit performance. The simulated data have the same format as the uncalibrated ramp data that will be made available to JWST observers, and include all metadata required for processing with the JWST Science Calibration pipeline. MIRISim was publicly released in 2018 as part of an Anaconda Python environment and is available at www.miricle.org.

1. Introduction

The Mid-Infrared Instrument (MIRI, Rieke et al. 2015) for the James Webb Space Telescope (JWST) provides measurements over the wavelength range of 5 to 28.5 μm . As the only mid-infrared instrument on JWST, its design (Wright et al. 2015) provides in a single package 4 scientific functions: photometric imaging in 9 filters between 5 and 27 μm over a 2.3 square arcminute field of view; coronagraphy in 4 filters between 10 and 27 μm , low spectral resolution ($R \sim 100$) slitted/slitless spectroscopy between 7 and 12 μm ; and medium spectral resolution ($R \sim 1300$ to 3700) integral field spectroscopy between 5 and 28.5 μm over fields of view up to 7.7 square arcseconds.

To support the testing of the hardware and the development of calibration software, a suite of MIRI simulators were created by the MIRI European Consortium. These simulators model the photon propagation through the various components of the instrument and produce detector images that are consistent with the expected on-orbit performance. Development is done in collaboration with the scientists and engineers who built and tested MIRI, and simulated observations are used for testing of commissioning and calibration pipeline software as well as for observation planning.

2. MIRISim design

MIRISim is a simulator package that combines a suite of simulators for individual MIRI components to be able to simulate the different modes of the instrument. The entire package has a Python-based object-oriented design and benefits from common (astronomy) utilities provided by the major scientific packages such as NumPy, SciPy, and Astropy. Below follows a short description of the key packages within MIRISim.

ObsSim is the main module of **MIRISim**; it provides the user-interface, handles input configuration files, sets up simulations as a series of dither and exposure events, and executes simulations. Internally, it acts as the main interface to the individual simulator modules, and handles the retrieval and writing of calibration files and output products.

SkySim is the **MIRISim** module that creates an astronomical “scene”: a description of the sky consisting of a series of e.g., a series of point or extended “sources” with positions, spatial extents, and spectral energy distributions (includes support for **PySynPhot** SEDs, or user-provided ASCII files). The users can optionally provide external FITS files that contain a spectral cube representing the sky for the mid-infrared wavelength range.

ImSim is the **MIRISim** module that simulates the light path through **MIRI** onto the Imager detector for the full-array and sub-array modes (excluding **LRS** and coronagraphy). It retrieves the astronomical scene from **SkySim** and produces an detector illumination model for the chosen full/sub-array, applying the instrument characteristics (distortion, spectro-photometric response, point-spread function). **LRSSim** is the **MIRISim** module that creates the illumination models for the two modes of the Low-Resolution Spectrometer (**LRS**), either focussing on one object in the slit for slitted mode, or dispersing the entire **LRS** field-of-view for slitless mode. For the Medium Resolution Spectrometer (**MRS**) mode, **ObsSim** queries **SkySim** to sample the astronomical scene onto an **MRS** appropriate spatial and spectral grid; the resultant “skycube” is stored as an intermediate product and provided as input into the **MRS** light path simulator **MIRI-MAISIE**. **MIRI-MAISIE** is based on the **MIRI SpecSim** simulator (Lorente et al. 2006), rewritten in Python using the Multipurpose Astronomical Instrument Simulator Environment (**MAISIE**; O’Brien et al. 2016). It produces detector illumination maps for the two **MRS** detectors, incorporating the imager slicer distortion, spectro-photometric response, and fringing effects. The detector illumination models from **ImSim**, **LRSSim**, or **MIRI-MAISIE** are stored by **ObsSim** as intermediate products.

The Sensor Chip Assembly simulator (**SCASim**; Beard et al. 2012) is the final step in **MIRISim**. It uses the input detector illumination information, and simulates the behaviour of the **MIRI** detectors to generate **MIRI** data files. It incorporates the major detector characteristics such as reference pixels, bad pixels, dark current, bias, gain, flatfield, non-linearity, noise sources (read-noise, Poisson noise), latency, and includes a cosmic ray event library.

3. Calibration and Data Models

The **MIRI** instrument characteristics are defined as a series of Calibration Data Products (**CDPs**), produced by the **MIRI** European Consortium, based on theoretical instrument models and/or data taken during multiple ground-based cryo-vacuum test campaigns. These **CDPs** represent the **MIRI** instrument team’s best working knowledge of **MIRI**, and have been a key deliverable to the Space Telescope Science Institute, where they are used in the calibration reference data system (**CRDS**) that supports the **JWST** Science Calibration Pipeline (Bushouse et al. 2017). During simulations, any requisite **MIRI** **CDPs** are automatically downloaded by **MIRISim**, and stored in a local cache for faster retrieval in subsequent simulations.

The structure of the input calibration files and the output data products are defined by a series of **MIRI** data models provided by a common **MIRI** utilities package called

MiriTE¹. These models are derived from the JWST data models that are both defined and used by the official JWST Science Calibration Pipeline (Bushouse et al. 2017). These models are used to ensure that the simulated data produced by MIRISim are compatible with the uncalibrated data that MIRI generates, and that these simulated data can be reduced and calibrated with the JWST Science Calibration Pipeline.

4. Distribution and usage

MIRISim was publicly released in 2018 as part of the MIRICLE software distribution, available for download at the MIRICLE website². The MIRICLE distribution is available for Linux or Mac, based on the Anaconda Python 3 distribution, and installs all requisite software and data file dependencies for MIRISim into a dedicated “mirisim” conda environment. After activating this conda environment, users can invoke “mirisim” (or “mirisim --help”) on the command line to start using MIRISim.

Users can design their simulations through a set of three configuration files (INI file format) that specify a) the simulation parameters (e.g., dither pattern, number and length of exposures, etc.), b) the astronomical scene (sources with positions and SEDs, type of background, etc.), and c) the advanced settings that control aspects of the simulator itself (e.g., whether to include certain steps such as bad pixels, cosmic rays, etc.). MIRISim can generate a default set of configuration files in-place, including a commented version that explains each parameter, with further explanation provided in the User’s Guide online. Alternatively, users can import MIRISim into their own Python session, and directly instantiate the simulation, scene, and simulator configuration objects, to allow for scripting of a series of simulations.

A User’s Guide, release notes, and Jupyter notebooks with walkthroughs for example simulation setups of each MIRISim mode are available on the MIRICLE website.

Acknowledgments. This presentation has been made possible thanks to the numerous scientists and engineers who contributed to the development and testing of MIRISim. MIRISim is developed by the MIRI European Consortium, as part of the JWST/MIRI consortium, which includes the following organizations: Ames Research Center, USA; Airbus Defence and Space, UK; CEA-Irfu, Saclay, France; Centre Spatial de Liège, Belgium; Consejo Superior de Investigaciones Científicas, Spain; Carl Zeiss Optronics, Germany; Chalmers University of Technology, Sweden; Danish Space Research Institute, Denmark; Dublin Institute for Advanced Studies, Ireland; European Space Agency, Netherlands; ETCA, Belgium; ETH Zurich, Switzerland; Goddard Space Flight Center, USA; Institut d’Astrophysique Spatiale, France; Instituto Nacional de Técnica Aeroespacial, Spain; Institute for Astronomy, Edinburgh, UK; Jet Propulsion Laboratory, USA; Laboratoire d’Astrophysique de Marseille (LAM), France; Leiden University, Netherlands; Lockheed Advanced Technology Center, USA; NOVA Opt-IR group at Dwingeloo, Netherlands; Northrop Grumman, USA; Max-Planck Institut für Astronomie (MPIA), Heidelberg, Germany; Laboratoire d’Etudes Spatiales et d’Instrumentation en Astrophysique (LESIA), France; Paul Scherrer Institut, Switzerland; Raytheon Vision Systems, USA; RUAG Aerospace, Switzerland;

¹github.com/JWST-MIRI/MiriTE

²miricle.org

Rutherford Appleton Laboratory (RAL Space), UK; Space Telescope Science Institute, USA; Toegepast Natuurwetenschappelijk Onderzoek (TNO-TPD), Netherlands; UK Astronomy Technology Centre, UK; University College London, UK; University of Amsterdam, Netherlands; University of Arizona, USA; University of Bern, Switzerland; University of Cardiff, UK; University of Cologne, Germany; University of Ghent; University of Groningen, Netherlands; University of Leicester, UK; University of Leuven, Belgium; University of Stockholm, Sweden; Utah State University, USA.

The MIRI project acknowledges the support from the following agencies: NASA; ESA; Belgian Science Policy Office; Centre Nationale D'Etudes Spatiales (CNES); Danish National Space Centre; Deutsches Zentrum für Luft- und Raumfahrt (DLR); Enterprise Ireland; Ministerio De Economia y Competividad; Netherlands Research School for Astronomy (NOVA); Netherlands Organisation for Scientific Research (NWO); Science and Technology Facilities Council; Swiss Space Office; Swedish National Space Board; UK Space Agency.

References

- Beard, S., Morin, J., Gastaud, R., Azzollini, R., Bouchet, P., Chaintreuil, S., Lahuis, F., Littlejohns, O., Nehme, C., & Pye, J. 2012, in *Astronomical Data Analysis Software and Systems XXI*, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461, 169
- Bushouse, H., Droettboom, M., & Greenfield, P. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512, 355
- Lorente, N. P. F., Glasse, A. C. H., & Wright, G. S. 2006, in *Astronomical Data Analysis Software and Systems XV*, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351, 61. astro-ph/0511036
- O'Brien, A., Beard, S., Geers, V., & Klaassen, P. 2016, in *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 991312
- Rieke, G. H., Wright, G. S., Böker, T., Bouwman, J., Colina, L., Glasse, A., Gordon, K. D., Greene, T. P., Güdel, M., Henning, T., Justanont, K., Lagage, P. O., Meixner, M. E., Nørgaard-Nielsen, H. U., Ray, T. P., Ressler, M. E., van Dishoeck, E. F., & Waelkens, C. 2015, *Publications of the Astronomical Society of the Pacific*, 127, 584
- Wright, G. S., Wright, D., Goodson, G. B., Rieke, G. H., Aitink-Kroes, G., Amiaux, J., Arichayanguas, A., Azzollini, R., Banks, K., Barrado-Navascues, D., Belenguer-Davila, T., Bloemmart, J. A. D. L., Bouchet, P., Brandl, B. R., Colina, L., Detre, Ö., Diaz-Catala, E., Eccleston, P., Friedman, S. D., García-Marín, M., Güdel, M., Glasse, A., Glauser, A. M., Greene, T. P., Groezinger, U., Grundy, T., Hastings, P., Henning, T., Hofferbert, R., Hunter, F., Jessen, N. C., Justanont, K., Karnik, A. R., Khorrami, M. A., Krause, O., Labiano, A., Lagage, P. O., Langer, U., Lemke, D., Lim, T., Lorenzo-Alvarez, J., Mazy, E., McGowan, N., Meixner, M. E., Morris, N., Morrison, J. E., Müller, F., rgaard-Nielson, H. U. N., Olofsson, G., O'Sullivan, B., Pel, J. W., Penanen, K., Petach, M. B., Pye, J. P., Ray, T. P., Renotte, E., Renouf, I., Ressler, M. E., Samara-Ratna, P., Scheithauer, S., Schneider, A., Shaughnessy, B., Stevenson, T., Sukhatme, K., Swinyard, B., Sykes, J., Thatcher, J., Tikkanen, T., van Dishoeck, E. F., Waelkens, C., Walker, H., Wells, M., & Zhender, A. 2015, *Publications of the Astronomical Society of the Pacific*, 127, 595

Preparing for JWST: A Detailed Simulation of a MOS Deep Field with NIRSpec

Giovanna Giardino,¹ Nina Bonaventura,² Jacopo Chevallard,³ Emma Curtis-Lake,³ Pierre Ferruit,¹ Peter Jakobsen,² Aurelien Jarno,⁴ Arlette Pecontal,⁴ Laure Piqueras,⁴ and The JADES Collaboration

¹*ESA, ESTEC, Noordwijk, The Netherlands; Giovanna.Giardino@esa.int*

²*The Cosmic Dawn Center, Copenhagen, Denmark*

³*Institut d'Astrophysique de Paris, Paris, France*

⁴*Université de Lyon, Univ. Lyon1, CNRS, CRAL Saint-Genis-Laval, France*

Abstract. JWST/NIRSpec will be the first multi-object spectrograph (MOS) to fly in space and it will enable the simultaneous measurement of up to ~ 200 spectra over the wavelength range $\sim 0.6 - 5.3 \mu\text{m}$, allowing us to study the rest-frame optical properties of large samples of galaxies out to $z \sim 9$, and the rest-frame UV out to $z > 10$.

To support the community in preparing NIRSpec MOS programs and getting ready to analyze the data, we present here a set of simulations closely mimicking the deep spectroscopic observations that will be performed as part of the JADES survey, a joint effort of the NIRCам and NIRSpec GTO teams. The simulations are made possible by the NIRSpec Instrument Performance Simulator software, a Fourier Optics wave propagation module coupled with a detailed model of the instruments optical geometry and radiometric response, and a detector simulator reproducing the noise properties and response of NIRSpec's two H2RG sensors. The targets for the simulations were selected from the JWST Extragalactic Mock Catalog, JAGUAR.

The simulation data package delivered here include more than 60 count-rate images corresponding to the exposures break-down of the low and medium resolution part of one of the two NIRSpec deep-field spectroscopic programs of the JADES survey. The simulated data consists of three dither pointings, for 4 different instrument configurations (low and medium resolution over the entire NIRSpec wavelength range), plus the extracted, background subtracted, spectral traces for each of the 370 targets and corresponding 2D-rectified spectra and calibrated 1D spectra, as well as the mock astronomical data used as the simulation input.

1. Introduction

To be prepared to exploit NIRSpec MOS data as soon as they will be acquired once JWST begins its science operations, it is crucial to have a good understanding of their properties and idiosyncrasies. In this work we present simulations of a NIRSpec observation of a deep field, realistically reproducing the instrument data for a set of exposures, including nodding and dither patterns. From the simulated exposures we extract mock NIRSpec-like data analogous to those that will be delivered at Stage 1 and 2 of the standard processing pipeline (JWST User Documentation 2016).

As template for the simulation we used one part of the NIRSpec Guaranteed Time Observation (GTO) program (see program GTO 1287¹). This program is part of the JWST Advanced Deep Extragalactic Survey (JADES), a joint project of the NIRCam and NIRSpec GTO teams and consists of a deep spectroscopic follow-up of high-redshift sources in the GOODS-South area identified by deep imaging observations with NIRCam. The deep-spectroscopy program consists of a 100 ks observation in instrument mode CLEAR/PRISM, that is spectral resolution ~ 100 over NIRSpec full wavelength range, plus 3×25 ks observations with instrument modes F070LP/G140M, F170LP/G235M, F290LP/G395M delivering medium resolution spectroscopy ($R \sim 1000$) over three NIRSpec bands (to cover the wavelength range $0.7 - 5.3 \mu\text{m}$) and 25 ks with instrument mode F290LP/G395H, $R \sim 3000$ over the wavelength range $2.9 - 5.3 \mu\text{m}$.

2. The IPS and the detector simulator

The NIRSpec Instrument Performance Simulator (IPS) was developed alongside NIRSpec by the Centre de Recherche Astrophysique de Lyon (CRAL) as part of a contract with Airbus Defence and Space (the prime contractor for NIRSpec). It is implemented in C++ with a Qt-based Graphical User Interface and runs on Linux systems. The software was initially developed to support the instrument design and assessment of performance-related trade-offs (Piqueras et al. 2008, 2010). Subsequently, it has been used to provide simulated NIRSpec exposures to prepare for the analysis of the data acquired during the instrument performance verification and calibration campaigns. Currently the IPS is used for simulating data to support the GTO programs and the astronomical community, and for testing and validation of the NIRSpec processing pipeline that is being developed at STScI.

The main component of the IPS is a Fourier Optics (FO) wave-propagation module that allows the incoming light wavefront to be followed through the main optical planes of the instrument taking into account the wavefront errors introduced by the different optical modules, the masking by the pupil and image-plane apertures, and the transmission efficiency of the individual elements. To do this the IPS contains a detailed geometrical and radiometric model of NIRSpec (Dorner et al. 2016) and the maps of wavefront errors of all the main optical modules. To convert the (noiseless) electron-rate maps generated by the FO module to count-rate images of individual exposures, equivalent to NIRSpec products as they will be delivered by the ‘ramp-to-slopes’ stage of STScI processing pipeline, we used a detector simulator package implemented in Python, that folds onto the computation the shot noise of the sources’ signal and the detector dark currents, readout noise, pixel-to-pixel cross talk and detector gain.

More details on the IPS and the detector simulator will be given in a future work.

3. The Astronomical Scene

The target sources for the simulation were selected from the fiducial mock catalog of the JADES extraGalactic UltraDeep Artificial Realizations (JAGUAR) package, developed

¹<https://jwst.stsci.edu/observing-programs/program-information>

by Williams et al. (2018). The catalog was generated using a novel phenomenological model for the evolution of galaxies and their properties, based on empirical constraints from current surveys between $0.2 < z < 10$. The model follows observed stellar mass functions, UV luminosity functions, integrated distributions including $M_{UV} - M_{\star}$, $\beta - M_{UV}$, and size-mass and size- M_{UV} distributions, and include galaxy SEDs thanks to self-consistent modeling of the stellar and nebular emission with the BEAGLE tool (Chevallard & Charlot 2016).

Like for the real NIRSpec MOS observations, sources have to be selected from the catalog and placed in the micro-apertures of the Micro-Shutter Assembly (MSA), according to considerations of target priority, maximizing the use of the MSA multiplexing capability and taking into account the observation dither pattern. In our case, to facilitate good local background subtraction, three adjacent microshutters in a column are opened for each target, and the source nodded between them during separate exposures. This 3-nodding pattern is repeated over three dithers, that is, pointing offsets of a few micro-shutters. To design the three MSA configurations, one per dither pointing, we used a custom planning tool developed by the GTO team that optimizes the MSA fine pointing so as to capture spectra of as many of the highest priority targets as possible. The planning algorithm selected a total of 370 targets over the three dithers, where 23% (including by construction the highest priority targets) are fully exposed and covered in all three dithers, 26% are 2/3-partially exposed and 51% are 1/3-exposed. Each dither pointing captures ~ 200 galaxies.

4. Simulated data and derived products

The astronomical scenes for the three dither and nodding positions provide the input for the IPS runs performed to generate the corresponding nine electron-rate images for each of the four instrument filter/disperser configurations. These were combined with the corresponding electron-rate maps for the three background scenes generated for the three MOS-masks of the dithers and used as input to the detector simulator to generate the count-rate maps for all the exposures planned in the GTO proposals and summarized in Table 1.

Table 1. Break-down of GTO program 1287 and of the simulation presented here, in terms of instrument filter/disperser configurations, number of dithers, number of nodding per dither, and number of exposures per nodding positions. The integration time of each exposure is 2801 s.

Config.	dithers	nodding	exposures	total n. exposures	total int. time (s)
CLEAR/PRISM	3	3	4	36	100,838
G140M/F070LP	3	3	1	9	25,210
G235M/F170LP	3	3	1	9	25,210
G395M/F290LP	3	3	1	9	25,210
G395H/F290LP*	3	3	1	9	25,210

*Simulations of this mode are not included in this data set.

Together with the count-rate image files we also deliver the extracted products: background subtracted traces, 2D rectified spectra and 1D-spectra, combined at nodding level. Although these products were generated with the GTO pipeline, the individual count-rate images and the products have a format similar to those that will be produced by Stage 1 and 2 of the official STScI pipeline. The processing steps of the official pipeline have been designed to closely match those of the GTO prototype.

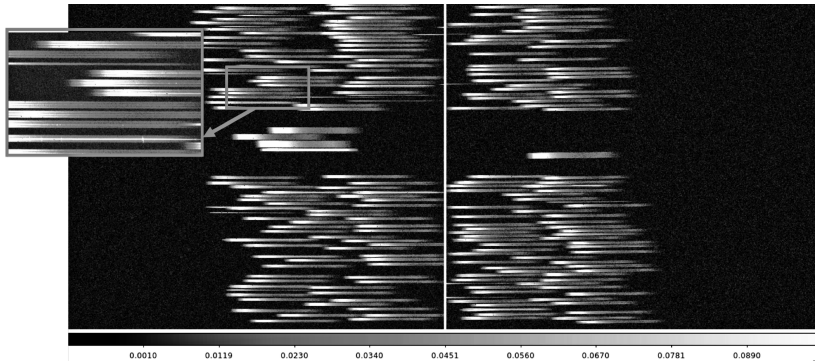


Figure 1. Count-rate image for one of the simulated prism exposures. In individual exposures, the signal in the 1×3 slitlets is dominated by the background emission.

5. Data download and future plans

All the data presented here are publicly available and can be downloaded from ESA web site: www.cosmos.esa.int/jwst-nirspec-simulations.

This first release of simulated data based on the JADES program clearly include some drastic simplifications, for instance all sources are assumed to be point-like. Moving forward, we plan to improve our simulation efforts and deliver ever more realistic simulated data and extracted products. Ultimately the data format of the simulated count-rate images will be made compatible for processing with the STScI official pipeline. Additionally we will also refine the data extraction and analysis process and, in a future work, we will describe the comparison between the input mock data and the high level products derived from the simulations (e.g., 1D-spectra and galaxies properties). To be kept up-to-date on new releases of NIRSpec simulated data you can register at: www.cosmos.esa.int/web/jwst-nirspec-simulations/register.

References

- Chevallard, J., & Charlot, S. 2016, MNRAS, 462, 1415
 Dorner, B., Giardino, G., Ferruit, P., et al. 2016, A&A, 592, A113
 JWST User Documentation 2016, Stages of Processing, STScI, <https://jwst-docs.stsci.edu>
 Piqueras, L., Legay, P., Legros, E., et al. 2008, in SPIE Conference Series, vol. 7017
 Piqueras, L., Legros, E., Pons, A., Legay, P., et al. 2010, in SPIE Conference Series, vol. 7738
 Williams, C. C., Curtis-Lake, E., Hainline, K. N., et al. 2018, Ap&SS, 236, 33

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

MegaPipe 2.0: 10000 Square Degrees of CFHT MegaCam Imaging

Stephen D.J. Gwyn

*Canadian Astronomy Data Centre, National Research Council, 5071 West
Saanich Road, Victoria, BC, V9E 2E7, Canada stephen.gwyn@nrc-cnrc.gc.ca*

Abstract. MegaPipe, the MegaCam data processing pipeline at the CADC, has been upgraded to version 2.0 and has processed over 10000 square degrees of the sky. MegaPipe has been operating since 2008. It was originally intended to increase the usage of archival MegaCam data by calibrating and stacking the images as they became public. That focus expanded to include processing data from the CFHT Large Programs such as the NGVS, OSSOS, VESTIGE and CFIS, as well as PI data. MegaPipe 2.0 represents several improvements. The advent of GAIA means that the astrometric calibration is considerably more accurate. The public release of Pan-STARRS allows photometric calibration of images even if they were taken under non-photometric conditions, by using the PS1 stars as in-field standards. Together this means that almost every MegaCam image can be astrometrically/photometrically calibrated to sufficient accuracy to allow stacking (30 mas and 0.01 magnitudes respectively). MegaPipe 2.0 also introduces an improvement to the stacking method. MegaPipe previously only stacked images that were centred on more or less the same part of the sky, which limited the number of images that could be stacked. MegaPipe 2.0 instead stacks on a grid of 10000x10000 pixel tiles, each half a degree square, evenly covering the whole sky. The result is that twice as much sky area can be stacked. There are now over 10000 square degrees of imaging in both the *ugriz* filters as well as the narrow band filters. The data is available for download at: <http://www.cadc-ccda.hia-ihc.nrc-cnrc.gc.ca/en/megapipe/access/graph.html>

1. Introduction

For the last decade, the MegaPipe pipeline has been processing data from the MegaCam wide-field mosaic camera on CFHT (Gwyn 2008). MegaPipe starts with individual MegaCam images, does a careful astrometric/photometric calibration on them, and then stacks them. It has been used extensively to process data from the CFHT Large Programs as well as data from numerous PI projects. The original intent, however, was to increase the utility of public archival MegaCam data. While this has been very successful, there were limitations on what archival data could be processed. MegaPipe 2.0 represents a major enhancement which improves the accuracy of the calibration and increases the number of images which can be processed.

2. Improved Astrometry

The advent of GAIA (Gaia Collaboration, Brown, A. G. A. et al. 2016) at once improves and simplifies the astrometric calibration of MegaCam data. Previously, individual

images could only be calibrated to around 0.1 arcsec using a combination of 2MASS and UCAC4. To achieve greater accuracy (as required for stacking and for tracking solar system objects), multiple overlapping MegaCam images had to be combined to produce a merged astrometric reference catalog. This reference catalog was then used to calibrate the individual images. However, individual images in isolation could not be calibrated with this method.

With the higher source density and greater accuracy of GAIA, individual images can be calibrated; there is no need for a merged astrometric reference catalog. Typical image-to-image residuals are 30 mas (previously 40 mas) or in dense stellar fields down to 10 mas (previously 20 mas).

The parameterization of the distortion has not changed. As before, rather than using the standard 2nd or 3rd order polynomial in (x, y) to convert to (RA, Dec) for each CCD of the mosaic, the distortion is instead measured as a polynomial in r^2 and r^4 for the whole mosaic, plus linear terms for the each CCD. This greatly reduces the number of parameters and decreases the likelihood of over-fitting.

3. Improved Photometric Calibration

The zero-point varies across the field of view of MegaCam. The previous version of MegaPipe dealt with this variation by computing a separate zero-point for each CCD of the mosaic. This correction was sufficient in general, as most of the CCDs have only small photometric variations across their area. The exception was the four corner CCDs of the mosaic which have typical variations on the order of 0.03 magnitudes. This zero-point variation is not static; it changes somewhat with each observing run and is different from filter to filter within an observing run.

MegaPipe 2.0 computes a map of the zero-point variations for each filter and observing run by cross matching with the Pan-STARRS catalog photometry (Magnier et al. 2016), transformed into the MegaCam photometric system. A set of 4×9 super-pixels is set up for each CCD. The photometric offsets are computed for each super-pixel by aggregating the offsets for every image taken during an observing run. Figure 1 shows an example of the zero-point variations.

The zero-point for each image as a whole is also set using the Pan-STARRS photometric catalog. This allows the calibration of data taken under non-photometric conditions and data taken on nights when insufficient photometric standards were observed.

The above procedure works well for every filter but the u-band. The Pan-STARRS photometry does not extend blue-wards of the g-band. One can in principle use the filter passbands and standard stellar spectra to create synthetic u-band photometry from the *grizy*-photometry. However, extrapolating in this manner is problematic. The u-band is sensitive to the metallicity-dependent line-blanketing effects in stars. Fields containing different metal-poor stars will end up having a different photometric calibration than fields with metal-rich stars. Consequently, the SDSS (Albareti et al. 2017) is used to calibrate the u-band data.

4. Tiling Scheme

The other change in MegaPipe 2.0 is how the images are stacked. Previously, MegaPipe only stacked images that were centred on more or less the same part of the sky. Since

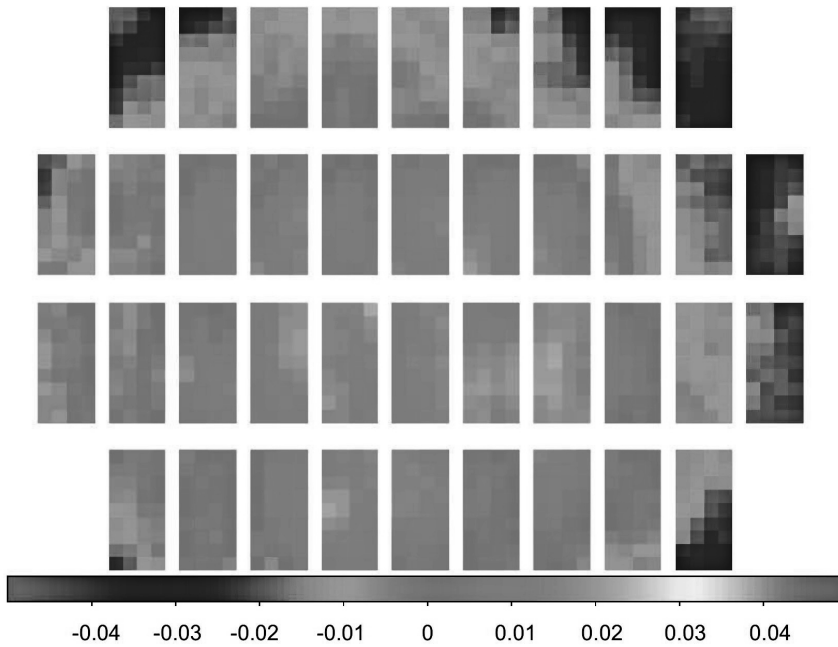


Figure 1. Zero-point variations across the MegaCam field of view for the CFHT MegaCam r filter, for the 17AM07 observing run

many MegaCam programs took their data with relatively small dithers, there was normally an obvious grid on which to stack the data. Images taken within 0.1 degrees of each other were grouped together using a friends-of-friends algorithm. However, this limited the number of images that could be stacked as not all program used small dithers. Further, as the archive grew larger, multiple programs started to overlap each other but in an inhomogeneous way.

MegaPipe 2.0 instead stacks on a set of tiles evenly covering the whole sky. The tessellation scheme is based on the Budavari rings¹. The centres of the tiles are spaced evenly by 0.5 degrees in declination and $0.5/\cos(\text{Dec})$ degrees in right ascension. The tiles are numbered 000-719 in RA (although every ring except for the equator will have fewer than 720 tiles) and 000-360 in Dec. The WCS of every tile is identical except for the values of CRVAL1 and CRVAL2. The tiles measure 10000×10000 pixels on a side, with the pixel scale being close to the MegaCam pixel scale of $0.185''/\text{pixel}$. This neatly provides approximately 3% overlap between adjacent tiles.

¹<https://dev.lsstcorp.org/trac/raw-attachment/ticket/2547/ps1.tessellation.tamas.pdf>

5. Conclusion

The improved astrometric and photometric methods of MegaPipe 2.0 mean that a far larger fraction of MegaCam archival images can be accurately calibrated. In particular, boot-strapping from the Pan-STARRS photometry means that data taken under non-photometric conditions can be calibrated. The tiling scheme means that even if images only overlap partially, they can still be stacked. The result is an improved calibration and a huge leap in coverage. 11457 square degrees of sky have been stacked by MegaPipe 2.0 as of October 2018. This is approximately double the coverage that was available a year previously, when the last batch of MegaPipe 1.0 processing was run. The coverage, of course, is not uniform. The depth and wavelength coverage is set by the heterogeneous programs MegaCam has been used for since 2003. Having a data set that is accurately calibrated and processed in a uniform way vastly increases the utility of the archival data.

Acknowledgments. Based on observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-France-Hawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institut National des Sciences de l'Univers of the Centre National de la Recherche Scientifique of France, and the University of Hawaii.

This work has made use of data from the European Space Agency (ESA) mission *Gaia* (<https://www.cosmos.esa.int/gaia>), processed by the *Gaia* Data Processing and Analysis Consortium (DPAC, <https://www.cosmos.esa.int/web/gaia/dpac/consortium>). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement.

References

- Albareti, F. D., Allende Prieto, C., Almeida, A., Anders, F., Anderson, S., Andrews, B. H., Aragón-Salamanca, A., Argudo-Fernández, M., Armengaud, E., Aubourg, E., & et al. 2017, *ApJS*, 233, 25. 1608.02013
- Gaia Collaboration, Brown, A. G. A. et al. 2016, *A&A*, 595, A2. 1609.04172
- Gwyn, S. D. J. 2008, *PASP*, 120, 212. 0710.0370
- Magnier, E. A., Schlafly, E. F., Finkbeiner, D. P., Tonry, J. L., Goldman, B., Röser, S., Schilbach, E., Chambers, K. C., Flewelling, H. A., Huber, M. E., Price, P. A., Sweeney, W. E., Waters, C. Z., Denneau, L., Draper, P., Hodapp, K. W., Jedicke, R., Kudritzki, R.-P., Metcalfe, N., Stubbs, C. W., & Wainscoat, R. J. 2016, *ArXiv e-prints*. 1612.05242

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Abstracting the Storage and Retrieval of Image Data at the LSST

Tim Jenness,¹ James F. Bosch,² Pim Schellart,² Kian-Tat Lim,³
 Andrei Salnikov,³ and Michelle Gower⁴

¹*Large Synoptic Survey Telescope, Tucson, AZ, USA; tjenness@lsst.org*

²*Princeton University, Princeton, NJ, USA*

³*SLAC National Accelerator Laboratory, Menlo Park, CA, USA*

⁴*National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign, IL, USA*

Abstract. Writing generic data processing pipelines requires that the algorithmic code does not ever have to know about data formats of files, or the locations of those files. At LSST we have a software system known as “the Data Butler,” that abstracts these details from the software developer. Scientists can specify the dataset they want in terms they understand, such as filter, observation identifier, date of observation, and instrument name, and the Butler translates that to one or more files which are read and returned to them as a single Python object. Conversely, once they have created a new dataset they can give it back to the Butler, with a label describing its new status, and the Butler can write it in whatever format it has been configured to use. All configuration is in YAML and supports standard defaults whilst allowing overrides.

1. Introduction

The Large Synoptic Survey Telescope (Ivezić et al. 2008), being built on Cerro Pachón in Chile, will be an automated astronomical survey system that will survey approximately 10,000 deg² of the sky every few nights in six optical bands. The associated Data Management System (Jurić et al. 2017; O’Mullane & LSST Data Management Team 2018) is required to process the data from this telescope and publish them as nightly alerts and as annual data releases. The LSST science pipelines (see for example Bosch et al. 2018; Bosch et al. 2019) have been designed such that the algorithmic code is insulated from having to know where data come from and how they have been serialized. The Butler is the system mediating the storage and retrieval of data, converting Python objects to data files and data files back to Python objects.

2. Butler Components

The Butler consists of a high-level Python API, and three core components: Schema, Registry access, and Datastore. The relationship of these components is shown in Fig. 1. The Schema defines the data model for relating datasets to each other and is defined consistently for all datasets and instruments. The Registry classes allow the

data model to be queried and are configurable via plugins to allow different backend database systems to be used. Finally the Datastore deals with the reading and writing of datasets themselves. Currently there are datastores for a POSIX file system, an in-memory cache, and chained datastores (where writes go to all datastores and reads pull from the first datastore to return it). For example, the Datastore configuration system allows certain dataset types to be cached in memory only and not written to a file system. This allows pipeline tasks to be linked together using the Butler as an intermediary without always incurring a write overhead. The configuration system allows the user to switch to writing intermediate files instead of caching them whilst debugging without changing any code. To support the Datastores, “Formatters” have to be written to serialize and deserialize Python objects to a variety of configurable data formats.

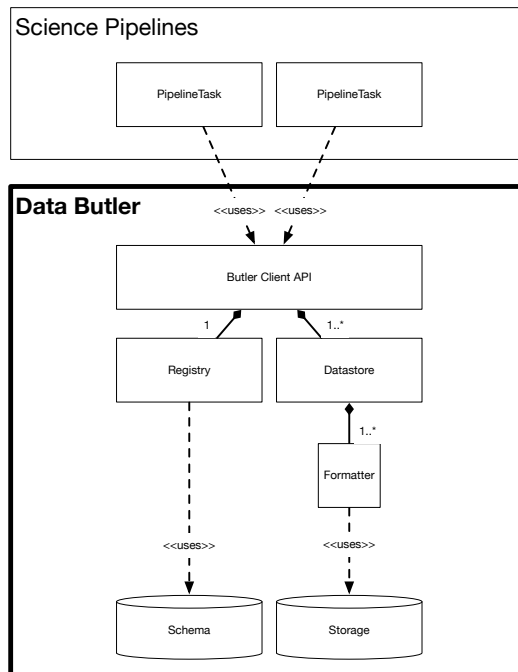


Figure 1. Architectural diagram of Butler components. Datastores use formatters to read from and write to storage, and the registry defines a schema that can be implemented in any database system. An example of the API is shown in §4.

3. Data Model

The Butler data model is designed to reflect the relationships between observations and calibrations, and also how the sky can be segmented into different regions, associating each dataset with a particular sky region. This allows you to ask which datasets are needed to calibrate another dataset, which datasets were taken with this filter between these dates, or which datasets would be needed to make a coadd covering this patch of

sky. It can also answer provenance queries, such as asking which coadds in a particular filter had at least this number of observations contributing. We are designing the Schema to be generally applicable for astronomical data and we are taking into account that we would like to map our schema to ObsCore (Louys et al. 2017) and CAOM-2 (Dowler 2012) data models in the future.

4. Using the Butler

Individual pipeline tasks work with Python objects. They put datasets into and retrieve datasets from the Datastores. The Butler maps a Python object to a serialization format through a “StorageClass” defined in the YAML configuration files for each Datastore. Changing the serialization format from FITS to HDF5 does not require any code changes for the user and is as simple as editing one line in the configuration file. Pre-defined components of a dataset, such as the WCS solution, can be retrieved without reading the full dataset if supported by the formatter. The components supported by each dataset type are defined at the StorageClass level, with code having to be written to assemble a Python composite object from the components and to disassemble a Python object into components.

Below is some user code for retrieving a raw HSC observation along with the relevant flatfield, processing it in some way, and then storing a new version with a different dataset type name. Calibration datasets can be retrieved by knowing the dataset that is to be calibrated.

```
from lsst.daf.butler import Butler

# Configure a new butler
butler = Butler("config.yaml")

# Specify the requested observation via metadata
dataId = {"instrument": "HSC", "obsid": "HSCA04090000"}

# Retrieve the raw data, process it, and store with new label
raw = butler.get("raw", dataId)
flat = butler.get("flat", dataId)
new = doSomething(raw, flat)
butler.put(new, "newlabel", dataId)

# Get just the WCS without reading the full dataset
wcs = butler.get("newlabel.wcs", dataId)
```

5. Header Translation

To be able to ingest instrument data into a Butler repository, the Butler has to understand some properties of the instrument including filters, detector information, and how to extract metadata from data headers. We have written a separate Python package, `astro_metadata_translator`, to support header translation and metadata extraction for astronomical instrument headers. The design of this new package has been influenced by the header translator written for ORAC-DR (Jenness & Economou 2015) and unifies the translation systems previously in use at LSST. New translators must be written to allow the Butler to understand data during ingest. Currently, translators exist for

DECam, CFHT MegaPrime, and SuprimeCam and Hyper-SuprimeCam from Subaru, with support for LSST test data being added as needed. This package solely depends on Astropy (Astropy Collaboration 2018) and does not need any LSST infrastructure.

```
from astropy.io import fits
from astro_metadata_translator import ObservationInfo

hdul = fits.open("hsc.fits")
obsInfo = ObservationInfo(hdul[0].header)
print(f"instrument={obsInfo.instrument}, "
      f"date-obs={obsInfo.datetime_begin}")
```

6. Summary

The Butler frees you from the worry of file formats and file systems when your main concern is processing and characterizing datasets. The Butler system is not LSST-specific, is written entirely in Python 3 (requiring Python 3.6 or newer following the project baseline (Jenness 2019)), and is driven by external configuration to suit different use cases. The Butler, currently undergoing heavy development and considered to be pre-beta, will be released at the end of 2018 alongside v17.0 of the LSST Science Pipelines. The source code for the Butler can be found at https://github.com/lsst/daf_butler, and the source code for the header translator can be found at https://github.com/lsst/astro_metadata_translator.

Acknowledgments. We thank Simon Krughoff, Gregory Dubois-Felsmann, and Meredith Rawls for their reviews of the draft paper. This material is based upon work supported in part by the National Science Foundation through Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.

References

- Astropy Collaboration 2018, *AJ*, 156, 123
- Bosch, J., et al. 2018, *Publications of the Astronomical Society of Japan*, 70, S5
- Bosch, J. F., et al. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 521
- Dowler, P. 2012, in *Astronomical Data Analysis Software and Systems XXI*, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461 of ASP Conf. Ser., 339
- Ivezić, Ž., et al. 2008, *ArXiv e-prints*, arXiv:0805.2366
- Jenness, T. 2019, in *ADASS XXVII*, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 541
- Jenness, T., & Economou, F. 2015, *Astronomy and Computing*, 9, 40. arXiv:1410.7509
- Jurić, M., et al. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of ASP Conf. Ser., 279
- Louys, M., et al. 2017, *Observation Data Model Core Components, its Implementation in the Table Access Protocol Version 1.1*, Tech. rep.
- O’Mullane, W., & LSST Data Management Team 2018, vol. 231 of *American Astronomical Society Meeting Abstracts*, 362.10

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

VO for Everyone - Getting Ready for the 4th ASTERICS DADI VO School

K. A. Lutz,¹ A. Nebot,¹ M. G. Allen,¹ and S. Derriere¹

¹*Observatoire Astronomique de Strasbourg, Université de Strasbourg, CNRS, UMR 7550, 67000 Strasbourg, France; research@katha-lutz.de*

Abstract. We present an update on the efforts of the European Virtual Observatory (EURO-VO) to inform and educate the astronomical community about Virtual Observatory (VO) tools and services. From the 20th to the 22nd of November 2018 the fourth and last Virtual Observatory school of work package 4 (DADI) of the European ASTERICS project has been taking place in Strasbourg. In the light of this event, this paper provides details on the VO-School highlighting the elements of the schools that we consider makes them a success. We present a short overview of recent developments, the current status and content, and future plans for tutorials on the Virtual Observatory.

1. Introduction

The Astronomy ESFRI & Research Infrastructure Cluster (ASTERICS) project is a Horizon 2020 project funded by the European Commission. ASTERICS aims to bring together researchers, scientists, engineers, hardware and software specialists from astronomy, astrophysics and astro-particle physics to tackle the challenges of transferring, processing and storing of large amounts of data. The fourth work package of ASTERICS focuses on Data Access, Discovery and Interoperability (DADI), which includes a yearly Virtual Observatory (VO) school for early career researchers (ECR) throughout the duration of the ASTERICS project. The aim of these VO-schools is to familiarise young researchers with various tools and services of the VO. After the schools ECRs are expected to both be able to use the VO efficiently for their research and act as ambassadors for the VO. Previous schools have been organised in Madrid (Spain) in November 2017 and December 2015, and in Strasbourg (France) in November 2016. The next and last ASTERICS VO-school is taking place from 20th to the 22nd of November 2018 again in Strasbourg. The program of this school can be found on the school webpage¹ and will be discussed further below.

2. Current Status of VO tutorials

After each VO-school, the updated tutorials are published on the EURO-VO webpage². Between schools, these tutorials are furthermore updated to account for recent ad-

¹<https://www.ASTERICS2020.eu/dokuwiki/doku.php?id=open:wp4:school4:program>

²<http://www.euro-vo.org/?q=science/scientific-tutorials>

vances in software development. In addition, comments submitted by participants via an anonymous feedback form at the end of each school was very useful to evolve and update tutorials. The tutorials on the EURO-VO webpage cover the following software packages, tools, and services:

- The CDS portal is an entry point to the services provided by the CDS and links to other (CDS) services. These services include ALADIN, SIMBAD and VIZIER. ALADIN is an interactive sky atlas allowing users to search, retrieve, and manipulate image, coverage and table data from the VO (Bonnarel et al. 2000; Boch & Fernique 2014). SIMBAD is the CDS object data base (Wenger et al. 2000, Loup et al. in prep.). VIZIER is a database of catalogues as collected and curated by the CDS (Ochsenbein et al. 2000).
- TOPCAT (Taylor 2005) and STILTS (Taylor 2006) are versatile table manipulation tools, which can also connect to the VO. While TOPCAT has a graphical user interface, STILTS is a command line tool and is thus able to handle bigger tables.
- More specialised tools that help to obtain and analyse certain data products are also included in the tutorials. CASSIS³ and SPLAT⁴ allow users to visualise and take measurements from spectra. VOSA analyses the spectral energy distribution (SED) of stars (Bayo et al. 2008).

In the tutorials, these tools are usually presented in the context of a science use case, e. g. identification of stellar clusters in the *Gaia* catalogues, obtaining information on particular galaxies or locating gravitational wave sources. The tutorials have been re-worked for the VO-school. These updates account for changes both in software tools (e. g. changes in the workflow that came about with the release of Aladin version 10) and publicly available data (e. g. release of *Gaia* DR2).

3. Fourth ASTERICS Virtual Observatory School

The programs of the VO-schools held in the ASTERICS project have evolved over time, taking into account the changes in the tools and services, and also the development of new scientific topics such as gravitational wave science. Feedback from participants of previous schools has also helped us to find a programme structure that best suits the needs of the participants in a three day event: The first two days focus on tutorials, while the third day is reserved for the students' own research project. On this third day, participants tackle their scientific questions with support from applying the skills they acquired the preceding days. To optimise the outcomes of day three, we interacted with participants to gather information on their scientific expertise and their previous experience with VO tools and services.

This school started with introductory talks about the VO and ASTERICS, which were followed by three tutorials on the first day. On the second day participants were guided through another three tutorials of about the same length. Then the second day finished with the "Treasure Hunt". In this game, five questions like "How many Novae

³<http://cassis.irap.omp.eu>

⁴<http://star-www.dur.ac.uk/~pdraper/splat/splat.html>

have been detected within 2 arcmin of the centre of M 31 to date?" are to be answered by the participants within a given amount of time and by using VO tools.

The tutorials selected and updated for the VO-School are:

"The CDS tutorial", which guides participants through the various CDS services and tools in the search for information on peculiar galaxies. At this VO-school it came with an new section on using Jupyter notebooks for access to CDS services.

"Determination of stellar physical parameters using VOSA", in which participants learn to use VOSA to assemble and analyse SEDs of stars.

"Accessing and cross matching big datasets with ADQL", an extensive beginners guide to ADQL.

"Electromagnetic follow-up of gravitational-wave events", which teaches how to use ALADIN to analyse the location of gravitational waves and plan follow-up observations.

"Exploring Gaia with TopCAT and STILTS", where participants learn to use TOPCAT and STILTS efficiently to find and analyse stellar clusters in *Gaia* DR2 catalogues.

"Advanced usage of HiPS and MOCs" in which participants create their own HiPS and subsequently find all sources in a *Gaia-WISE* cross-match that are located within their HiPS and at low Galactic extinction.

Based on the experience of the series of VO-schools in ASTERICS, we can clearly identify that interaction between the participants and VO expert tutors (both scientists and software developers) is a key element for the success of these schools. A common observation is that the tutorials enable a much deeper understanding of the capabilities of the tools and what the VO can offer. While information is available on-line, the personal interaction afforded in a school greatly enhances the transfer of knowledge.

4. Future Plans

Python⁵ is becoming more widely used in astronomy data analysis. We therefore work towards including example Python workflows in our tutorials. We consider interactive Jupyter notebooks (Kluyver et al. 2016) as a suitable way to present these workflows. A Python package of interest for these tutorials is **Astropy** (Astropy Collaboration et al. 2013), which aims to provide core utilities for astronomers. There are two Astropy affiliated packages that allow to query VO-services and will thus be relevant for the tutorials: **pyVO**⁶ and **astroquery**⁷. In addition we would like to familiarise astronomers with **MOCpy**⁸, a package to create, modify and use multi-order coverage (MOC) maps, and **ipyaladin**⁹, which provides a way to display an Aladin Lite widget in Jupyter notebooks. One example for these Jupyter notebooks was presented in the tutorial "All-sky astronomy with HiPS and MOCs"¹⁰ (Derriere 2019). This pre-ADASS tutorial was

⁵<https://www.python.org/>

⁶<https://pyvo.readthedocs.io/en/latest/>

⁷<https://astroquery.readthedocs.io/en/latest/>

⁸<https://mocpy.readthedocs.io/en/latest/>

⁹<https://github.com/cds-astro/ipyaladin>

¹⁰<http://cds.unistra.fr/adass2018/>

also live streamed to YouTube. There, the CDS also publishes short video tutorials on VO-related topics on the CDS YouTube channel¹¹.

5. Summary and Conclusion

Between the ADASS XXVIII meeting and the publication of this paper, the Fourth ASTERICS VO-School was successfully held. In a final, anonymous evaluation round, participants found the knowledge they acquired throughout the school very useful. However, they also requested that more examples for scripting and automating certain tasks should be shown and taught. Bringing Python to upcoming VO-Schools and taking more time to discuss tools as STILTS and the scripting mode of ALADIN will satisfy this need and help to prepare astronomers for the era of data-intensive astronomy.

Acknowledgments. The authors would like to thank the tutors and the LOC of the Fourth ASTERICS VO-School for their dedication and help to make the school run successfully. We also thank the participants and encourage their future roles to spread the knowledge obtained in the school.

References

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., Unther, H. M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J. E. H., Singer, L., Fox, R., Weaver, B. A., Zabalza, V., Edwards, Z. I., Azalee Bostroem, K., Burke, D. J., Casey, A. R., Crawford, S. M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P. L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., & Streicher, O. 2013, *A&A*, 558, A33. 1307.6212
- Bayo, A., Rodrigo, C., Barrado Y Navascués, D., Solano, E., Gutiérrez, R., Morales-Calderón, M., & Allard, F. 2008, *A&A*, 492, 277. 0808.0270
- Boch, T., & Fernique, P. 2014, in *Astronomical Data Analysis Software and Systems XXIII*, edited by N. Manset, & P. Forshay, vol. 485 of *Astronomical Society of the Pacific Conference Series*, 277
- Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. 2000, *A&AS*, 143, 33
- Derriere, S. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of *ASP Conf. Ser.*, 685
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., et al. 2016, in *ELPUB*, 87
- Ochsenbein, F., Bauer, P., & Marcout, J. 2000, *Astronomy and Astrophysics Supplement Series*, 143, 23. astro-ph/0002122
- Taylor, M. B. 2005, in *Astronomical Data Analysis Software and Systems XIV*, edited by P. Shopbell, M. Britton, & R. Ebert, vol. 347, 29
- 2006, in *Astronomical Data Analysis Software and Systems XV*, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of *Astronomical Society of the Pacific Conference Series*, 666
- Wenger, M., Ochsenbein, F., Egret, D., Dubois, P., Bonnarel, F., Borde, S., Genova, F., Jasiewicz, G., Laloë, S., Lesteven, S., & Monier, R. 2000, *A&AS*, 143, 9. astro-ph/0002110

¹¹https://www.youtube.com/channel/UCUESQ17rNupLlV_VcceE0Ng

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

ALiX: An Advanced Search Interface for Aladin Lite

Laurent Michel,¹ Thomas Boch,¹ Xinyu Shan,² and Jie Wang²

¹*Université de Strasbourg - CNRS - Observatoire astronomique, Strasbourg, France; laurent.michel@astro.unistra.fr*

²*Université Technologique de Belfort Montbéliard, Belfort, France*

Abstract. ALiX is a flexible catalog portal based on Aladin Lite. It is designed to use an interactive sky view as a primary selection tool. The ALiX view is constantly updated with data queried in the host database. It offers advanced functionalities to allow mixing local data with VO data. Users can plot by hand areas of interest and manage an history of the views. ALiX has no dependency on any specific data source; it can be integrated with existing portal.

1. Introduction

After many years of discussions, leading to consensus, sometimes difficult to reach, and thanks to the perseverance of a community of motivated people, the VO technologies are now widely used by astronomers. Thousands of resources are immediately available from tools such as Aladin or Topcat, or from software frameworks such as AstroQuery. Major agencies are basing their archives on VO protocols (ESASky (Merín et al. 2017), ESO Science Archive (Romaniello et al. 2018), CADC (Redman & Dowler 2013)). A particular set of VO technologies issued from the HEALPix sky tessellation allows lightweight clients to browse very large data sets (HiPS image surveys, progressive catalogs) and to describe their sky coverage in a very efficient way (MOC (Fernique et al. 2014)). Aladin Lite (Boch & Fernique 2014), one of the most popular of these tools, offers a very simple way to incorporate a VO portal into Web pages. The other side of the coin is the difficulty of retrieving particular services among thousands of others. ALiX is a widget, based on Aladin Lite, which can be connected on both local data source and VO world thanks to an advanced facility for discovering VO resources.

2. Overview

ALiX (Figure 1) can be easily embedded in a Web portal. It allows combinations of data searched in the local database with VO HiPS (images or catalogs) (Fernique et al. 2017). The local data source is named master resource. The display of the master resource is automatically updated with the Aladin Lite view changes. A reference position can be set to help users to come back home after a sky exploration. The background image is chosen among all HiPS surveys. It can be changed at any time.

The sky view can be overlaid with sources from any catalog available on the MOC server (Boch & Fernique 2018) in addition to the 2 main identification catalogs: NED

and Simbad. By construction, the search radius covers the whole view for progressive catalogs. It is limited to 1 degree otherwise. The master resource can be declared as a progressive catalog if the backend server can feed up ALiX quickly enough or if the query is setup to select a limited number of sources sorted by some physical criteria.

ALiX is written in Javascript/JQuery. JQuery-ui components are also used as well as Bootstrap CSS. Some of the Aladin Lite functions had to be overridden in a specific module. For this reason, the ALiX distribution contains a frozen version of Aladin Lite.

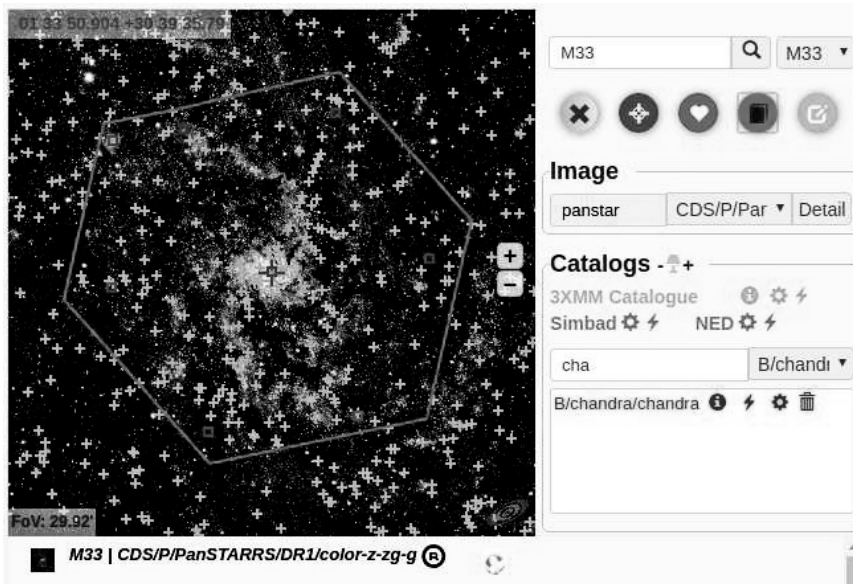


Figure 1. ALiX screenshot with a polygon drawn by the user and a bookmarked view.

3. Features

3.1. HiPS Resource Selection

The selection of HiPS resources relies on the very short response delay of the MOC server. The list of available HiPS covering the current view and having meta data matching the query is updated as long as the user is typing his request. He/she has just to click on that list to display the corresponding survey (Figure 2).

3.2. Bookmark Facility

The current view can be bookmarked at any time. Bookmarked views can not only be restored with all the original features but also viewed as PNG images. Bookmarks can also be annotated with text.

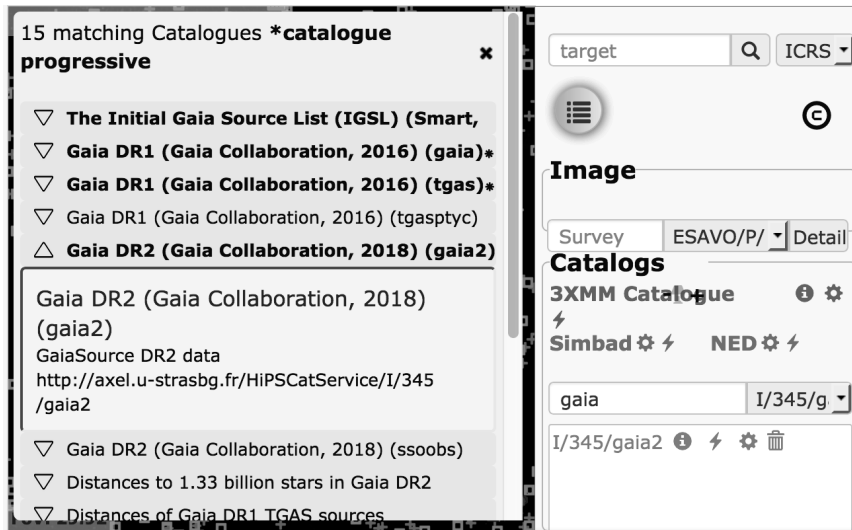


Figure 2. List of catalogs proposed by the MOC server and matching the keyword "gaia". Progressive catalogs are in boldface.

3.3. Region Editor

Polygonal regions can be plotted over the view. A user handler is called when the polygon is closed. This feature can be used to set up searches by regions or cutout services. In addition, a footprint can be drawn by an API callback. This might be used to visualize cutout limits.

3.4. Configuration

ALiX instances are configured by a Javascript object where many parameters and behaviours are defined.

The default view parameters such as the reference position, the view size and the default survey are set that way.

The master resource can also be tuned by this configuration object. Its data source is defined by a templated URL. Users can provide handlers called when sources are selected or deselected. This feature can be used to search additional data attached to the clicked source. It has been tested on our new 3XMM catalog interface (Michel et al. 2015) which provides that way an easy access to many datasets (spectra, timeseries, images ...) associated to XMM sources selected on the ALiX view.

3.5. Public API

The public API exposed by ALiX allows the host application to dynamically change the reference position or the master resource. It also provides more advanced features such as the possibility of hiding sources not matching given parameters (magnitude, velocity ...)

4. Status and Prospects

ALiX is published on GitHub¹ under MIT license. The GitHub project does include documentation and lots of examples. An ALiX based interface for browsing XMM-Newton sources has been published² at the same time. After these concrete implementations, we plan to use ALiX to explore TAP resources. This feature has been tested on various services and works very well as long as TAP services support the query bursts generated by rapid view moves. ALiX will likely be integrated into TapHandle in 2019.

Acknowledgments. We would like to thank the CDS for getting involved in that development and for having adapted the MOC server for our requirements. We would also thank the XMM-Newton Survey Science Consortium which funded that work. Our final thank goes to the whole VO community which always pays attention to this kind of project.

References

- Boch, T., & Fernique, P. 2014, in *Astronomical Data Analysis Software and Systems XXIII*, edited by N. Manset, & P. Forshay, vol. 485 of *Astronomical Society of the Pacific Conference Series*, 277
- 2017 (accessed 19-November-2018), MocServer. URL <http://alasky.unistra.fr/MocServer/query>
- Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017, HiPS - Hierarchical Progressive Survey Version 1.0, IVOA Recommendation 19 May 2017. 1708.09704
- Fernique, P., Boch, T., Donaldson, T., Durand, D., O'Mullane, W., Reinecke, M., & Taylor, M. 2014, MOC - HEALPix Multi-Order Coverage map Version 1.0, IVOA Recommendation 02 June 2014. 1505.02937
- Merín, B., Salgado, J., Giordano, F., Baines, D., Sarmiento, M.-H., Martí, B. L., Racero, E., Gutiérrez, R., Pollock, A., Rosa, M., Castellanos, J., González, J., León, I., Ortiz de Landaluce, I., de Teodoro, P., Nieto, S., Lennon, D. J., Arviset, C., de Marchi, G., & O'Mullane, W. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of *Astronomical Society of the Pacific Conference Series*, 495
- Michel, L., Grisé, F., Motch, C., & Gomez-Moran, A. N. 2015, in *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*, edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of *Astronomical Society of the Pacific Conference Series*, 173
- Redman, R. O., & Dowler, P. 2013, in *Astronomical Data Analysis Software and Systems XXII*, edited by D. N. Friedel, vol. 475 of *Astronomical Society of the Pacific Conference Series*, 159
- Romaniello, M., Zampieri, S., Delmotte, N., Forchì, V., Hainaut, O., Micol, A., Retzlaff, J., Vera, I., Fourniol, N., Khan, M. A., Lange, U., Sisodia, D., Stellert, M., Stoehr, F., Arnaboldi, M., Spiniello, C., Mascetti, L., & Sterzik, M. F. 2018, *The Messenger*, 172, 2

¹<https://github.com/lmichel/alix>

²<http://xcatdb.unistra.fr/3xmmdr8>

Session XIV

Tutorials

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

All-Sky Astronomy with HiPS and MOCs

Sébastien Derriere¹

¹*Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg,
 UMR 7550, F-67000 Strasbourg, France;
 sebastien.derriere@astro.unistra.fr*

Abstract. This tutorial intended to teach participants how to use recent Virtual Observatory standards allowing exploration and querying of all-sky datasets. The Hierarchical Progressive Survey (HiPS) and the Multi-Order Coverage map (MOC) can be used by data providers to expose their datasets (images or catalogs), and astronomers can use them to perform complex queries on all-sky datasets. Participants have created image and catalog HiPS, learned how to compare them to reference datasets, and share them in a web page. Advanced usage with the Table Access Protocol (TAP) and with the Python libraries Astropy and MOCPy was also shown.

1. Introduction

After a short introduction presenting the concepts of HiPS (Fernique et al. 2015, 2017a) and MOC (Fernique et al. 2014a), and making sure the test data samples were available to everyone, participants were invited to follow the instructions¹ during the 2 hours tutorial session:

1. Generate a HiPS image survey and its associated MOC;
2. Generate a catalog HiPS;
3. Compare these HiPS with other surveys;
4. Query a catalog by MOC;
5. Publish the HiPS and share them with Aladin Lite;
6. Perform advanced queries with TAP/ADQL, and Python.

2. Generate HiPS image survey and MOC

Aladin Desktop (Bonnarel et al. 2000; Boch et al. 2011) can be used to generate a HiPS image survey from a collection of calibrated FITS images. A small sample of images from the ISOPHOT 170 micron Serendipity Survey (Stickel et al. 2007), covering a fraction of the galactic plane was used for this step.

¹<http://cds.unistra.fr/adass2018/>

2.1. HiPS image

The first step of the tutorial was to compute a HiPS from the sample ISOPHOT images. In Aladin Desktop, the HiPS generation program (Fernique et al. 2014b) is available from the menu **Tool > Generate a HiPS based on... > An image collection (FITS, JPEG or PNG)**. Providing the directory containing the source files, and the Target directory where the hierarchy of files of the HiPS will be stored, the HiPS is quickly computed, and empty pixel values (0 in our case) can be treated so they appear transparent in the output files. PNG version of the HiPS image tiles are computed in addition to the FITS ones, in order to display the HiPS in Aladin Lite.

The image HiPS (computed by default to a maximum resolution corresponding to Healpix order 10) can be viewed in Aladin Desktop, and compared to over 300 other HiPS image surveys available in the data collections.

2.2. MOC for an image survey

A MOC file has been automatically generated: the corresponding `MOC.fits` file can be visualized in Aladin Desktop. The default MOC has been computed at order 10, which can be checked by overlaying the Healpix grid (Alt+W). A more detailed MOC can be computed, for example at level 11, from the menu **Coverage > Generate a MOC based on > An image collection**.

The two MOCs and their properties can then be compared.

3. Catalog HiPS

The `Hipsgen-cat` tool² is used to compute the progressive HiPS version of catalogs.

The MSX6C Infrared point source catalog (Egan et al. 2003) has been used. The 431711 rows have been extracted in VOTable base64 format from VizieR and provided to the participants.

The catalog HiPS, with a maximum Healpix order 6, is generated by running the following command:

```
java -jar Hipsgen-cat.jar -cat MSX6C -in vizier_votable.b64 -f VOT
-ra _RAJ2000 -dec _DEJ2000 -lM 6 -score B2 -desc -out /path2/MSX6C
```

In the progressive catalog, the sources with the highest values of B2 will be shown first, and sources with lower B2 values will only appear when one zooms in. This progressive catalog can be opened as a local file in Aladin Desktop.

4. Comparing with other surveys

Participants were invited to compare the image and catalog HiPS with other reference surveys: images from PanSTARRS DR1 color, collection of HST images near the V band, and Herschel SPIRE color, and a HiPS version of the Gaia DR2 catalog. These various surveys can be compared by changing the opacity of the overlaid images, or using the multi-view and *match* buttons to align the views.

²<http://aladin.u-strasbg.fr/hips/HipsCat.gml>

5. Queries by MOC

One can perform logical operations on MOCs (such as union, intersection), or use MOCs to perform complex spatial queries. After computing a new MOC corresponding to the intersection of 1) the ISOPHOT MOC computed in section 2.1, 2) the HST V band MOC and 3) the Gaia MOC (Coverage > Logical operations), the Gaia DR2 catalog was queried by MOC, in order to retrieve the 4263 sources located inside the intersection.

6. Publishing with Aladin Lite

With Aladin Lite (Boch & Fernique 2014, 2017), any image HiPS can be easily embedded in a web page, if its directory structure is accessible on a web server.

The Aladin Lite API³ explains how to integrate custom image and catalog HiPS in Aladin Lite. Participants could create a local web page, integrating an Aladin Lite instance with HiPS stored on a remote server. Or they could start a local http server, for example running `python3 -m http.server`, and integrate the ISOPHOT HiPS and the progressive MSX6C catalog created in section 2 in a local web page.

7. Advanced usage of HiPS and MOCs

7.1. Healpix in TAP/ADQL queries

The goal of this section was to run a TAP query on the Gaia DR2 catalog to visualize the rotation pattern of the Large Magellanic Cloud. TAP queries (written in ADQL) on Gaia DR2 data can be done in different ways: from the TAP Vizier web interface, from TOPCAT, or directly from Aladin Desktop, by choosing the "*by criteria*" access method.

We make use of the Healpix sky partitioning to compute averaged values of position, proper motions, parallax and velocities for Gaia DR2 sources in different Healpix cells overlapping the LMC, using the following ADQL query (note the GROUP BY h).

```
SELECT avg(ra), avg(dec), avg(pmra), avg(pmdec), avg(parallax),
avg(radial_velocity), healpix(ra, dec, 8) AS h FROM "I/345/gaia2"
WHERE sqrt(pmra*pmra+pmdec*pmdec)<30 AND
CONTAINS( POINT('ICRS',ra,dec),
CIRCLE('ICRS',80.8942,-69.7561,2.5) ) = 1 GROUP BY h
```

The query is executed in asynchronous mode from Aladin Desktop, and visualized with a dedicated filter. When subtracting for the mean proper motion of stars in the area (which can also be found with an ADQL query), the rotation of the LMC is clearly visible.

7.2. Programmatic use in Python

There is a Python interface to the CDS MOCServer (Fernique et al. 2017b), a server containing MOC and metadata information for a large number of datasets. The MOC-

³<https://aladin.unistra.fr/AladinLite/doc/API/>

Server can be queried with the `query_region()` method of the `astroquery.cds` Python library⁴, and the `region` parameter can be a `mocpy.MOC` object. This allows to find all datasets intersecting any MOC.

Participants used a Jupyter notebook to do the following with Python commands:

- Download custom and official MOCs;
- Compute their intersection;
- Visualize them using `pyplot`;
- Find a selection of VizieR infrared catalogs containing data in the MOC that we have created in section 4;
- Visualize MOCs with Aladin Lite using `ipyaladin`, and overlay catalogs;
- Query a catalog by MOC.

Acknowledgments. The author would like to thank all the CDS staff members that helped assisting the participants during this tutorial.

References

- Boch, T., & Fernique, P. 2014, in *Astronomical Data Analysis Software and Systems XXIII*, edited by N. Manset, & P. Forshay, vol. 485, 277
- 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512, 105
- Boch, T., Oberto, A., Fernique, P., & Bonnarel, F. 2011, in *Astronomical Data Analysis Software and Systems XX*, edited by I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots, vol. 442, 683
- Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. 2000, *Astronomy and Astrophysics Supplement Series*, 143, 33
- Egan, M. P., Price, S. D., Kraemer, K. E., Mizuno, D. R., Carey, S. J., Wright, C. O., Engelke, C. W., Cohen, M., & Gugliotti, M. G. 2003, *VizieR Online Data Catalog*, V/114
- Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017a, *HiPS - Hierarchical Progressive Survey Version 1.0*, Tech. rep.
- Fernique, P., Allen, M. G., Boch, T., Oberto, A., Pineau, F. X., Durand, D., Bot, C., Cambrésy, L., Derriere, S., Genova, F., & Bonnarel, F. 2015, *A&A*, 578, A114
- Fernique, P., Boch, T., Donaldson, T., Durand, D., O’Mullane, W., Reinecke, M., & Taylor, M. 2014a, *MOC - HEALPix Multi-Order Coverage map Version 1.0*, Tech. rep.
- Fernique, P., Boch, T., Oberto, A., & Pineau, F. X. 2017b, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512, 133
- Fernique, P., Boch, T., Pineau, F., & Oberto, A. 2014b, in *Astronomical Data Analysis Software and Systems XXIII*, edited by N. Manset, & P. Forshay, vol. 485, 281
- Stickel, M., Krause, O., Klaas, U., & Lemke, D. 2007, *A&A*, 466, 1205

⁴<https://astroquery.readthedocs.io/en/latest/cds/cds.html>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Working with the Hubble Space Telescope Public Data on Amazon Web Services

Ivelina G. Momcheva

Space Telescope Science Institute, Baltimore, MD, USA; imomcheva@stsci.edu

Abstract. In May 2018 STScI announced that ~110 TB of archival observations from the *Hubble Space Telescope* are available in cloud storage on Amazon Web Services. This tutorial provides an introduction to accessing this dataset and to AWS cloud computing in general for users who have not previously used cloud resources. We demonstrate how to access the Hubble Public Dataset on AWS in order to carry out a variety of tasks. Participants are introduced to the `astroquery.mast` and `boto3` Python client libraries. We demonstrate operations with the data such as retrieving, displaying and running analysis on single images. We then show how to scale up analysis through serverless computing. Finally we browse some advanced capabilities such as logging, price estimation and machine learning. The tutorial is geared toward novice AWS users. Intermediate Python knowledge is strongly recommended.

1. Introduction

The *Hubble Space Telescope* has undeniably expanded our understanding of the universe during its 28 years in space so far, but this is not just due to its superior view from space. One of the major advantages to *Hubble* is that every single image it takes becomes public within six months (and in many cases immediately) after it is beamed back to Earth. The treasure trove that is the Hubble archive has produced just as many discoveries by scientists using the data “second hand” as it has from the original teams who requested the observations.

In May, 2018 we announced that 110 TB of *Hubble*’s archival observations are available in cloud storage on Amazon Web Services (AWS) which provides unlimited access to the data right next to a wide variety of computing resources. These data consist of all raw and processed observations from the currently active instruments: the Advanced Camera for Surveys (ACS), the Wide Field Camera 3 (WFC3), the Cosmic Origins Spectrograph (COS), the Space Telescope Imaging Spectrograph (STIS) and the Fine Guidance Sensors (FGS). The data on AWS¹ are kept up to date with the data held in the Mikulski Archive for Space Telescopes (MAST). New and reprocessed data are updated on AWS within 20 minutes of them being updated at MAST. The combination of cloud computing with one of the highest value dataset in astronomy has the potential to yield new scientific discoveries by allowing users to do large scale data analysis and utilize cloud services.

¹<https://registry.opendata.aws/hst/>

We have created a tutorial which demonstrates how to access the Hubble Public Dataset on AWS in order to carry out a variety of tasks. Access to the data is provided through a custom extension to the `astroquery` Python library – `astroquery.mast` – and the AWS client Python library `boto3`. The tutorial introduces participants to the basic functionality. It further demonstrates operations with the data such as retrieving, displaying and running analysis on single images. It then shows how to scale up the analysis to hundreds on images through AWS Lambda serverless computing.

2. Learning Objectives

The primary objective for the tutorial is to get users started with the *HST* AWS dataset. This goal will be accomplished with the following activities:

- Learn about the Hubble Public Dataset on AWS
- Learn about `astroquery.mast` and `boto3`
- Download the data
- Visualize the data and carry out data analysis
- Create a function that can be parallelized
- Run a function using Lambda (serverless computing)
- Basic exploration of machine learning resources

Even though cloud computing is now widely used in industry applications of big data, the uptake of this technology in the astronomical community has been slow. A secondary objective of the tutorial is to introduce the participants to several key services provided by cloud computing platforms including on-demand computational resources, storage, serverless compute and machine learning capabilities. Along with this, participants learn about estimating costs, logging activity and connecting different cloud services to execute a task.

The secondary learning objectives are accomplished with the following tasks:

- Creating an AWS account and logging in
- Starting Amazon machine images (AMIs) image and connecting to it
- Creating a Docker container based on a template
- Writing a Lambda function
- Different types of cloud storage, creating a new bucket, connecting to them
- Logging activity
- Prototyping computations and estimating cost

3. Tutorial

The tutorial is based on blog posts in the Mast Labs blog.² A basic/intermediate knowledge of Python is a prerequisite for this tutorial. Users need a laptop and an internet connection. The contents of the tutorial are hosted in a public Google document:

<https://tinyurl.com/adass2018-aws>

Figure 1 shows the final output from the Lambda task. Lambda is serverless compute, where the user specifies the software but not the hardware. It is well suited for small tasks that are easily parallelize-able. In our example we carry out source detection using `sep`³ and produce a gray-scale PNG file of the image with red circles at the positions of all sources. The example case executes in seconds on one hundred images and the cost is covered under the free Lambda tier.

Users are encouraged to work through the tutorial on their own time. Comments and suggestions for improvements are welcome at imomcheva@stsci.edu.

Even though AWS offers a free tier for most of its services, some of the tutorial requirements are not covered under it and a valid credit card is needed order to create an AWS account even if no charges are incurred. Going through the materials in the tutorial cost less than 5 cents to run and at the end of the tutorial we show users how to clean their workspace so no further charges are incurred. For larger scale computation AWS does offer cloud credits for research grants⁴ through a competitive process. Starting in HST proposal cycle 26 (2018), STScI established a new type of proposal specifically utilizing this dataset which provides up to \$10,000 for AWS services. Additionally, all archival and general observer programs can request AWS funds in their budgets.

4. Conclusion

The tutorial presented here introduces users to the *HST* data on AWS and to cloud computing in general. Cloud compute combined with astronomical data offers new capabilities for discovery. Some of the services offered by cloud providers (e.g., Lambda) do not have easy analogues that can be run locally and with so little effort. We envision many more use cases for cloud computing in the near future.

Acknowledgments. This tutorial was supported by AWS who provided cloud credits for all participants. We thank the tutorial helpers C. Brasseur, P.-L. Lim and N. Miles.

²<https://mast-labs.stsci.io/>

³<https://github.com/kbarbary/sep>, a Python wrapper around the core algorithms of Source Extractor

⁴<https://aws.amazon.com/grants/>

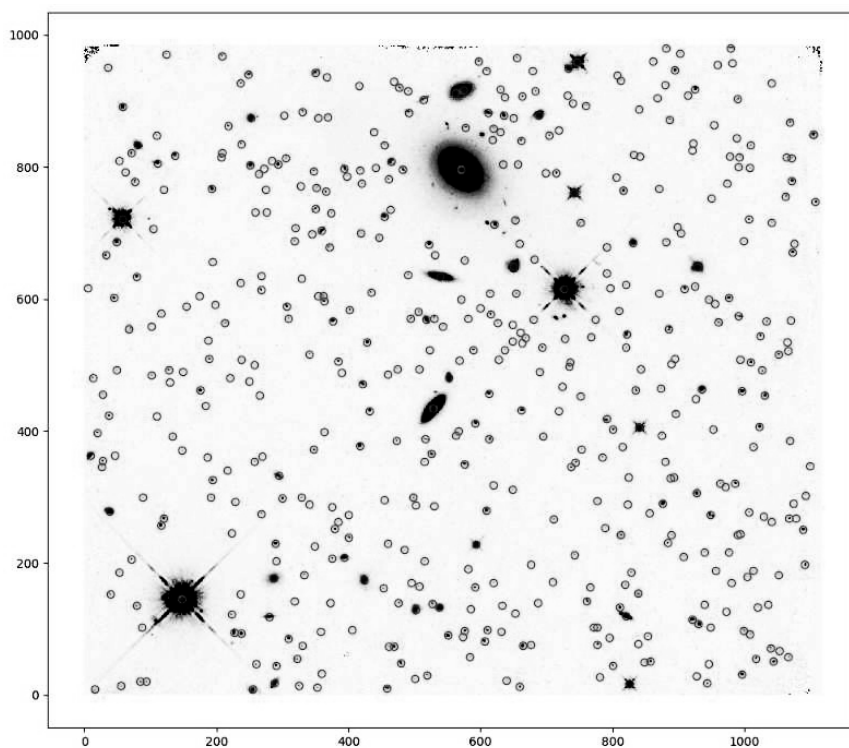


Figure 1. Sample output from the AWS Lambda function executed as part of the tutorial. The background is a grayscale *HST* WFC3/IR image. Red circles mark the position of detected sources.

Session XV

Focus Talks and Demo Booths

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Building LOFAR as a Service

A.P. Mechev,¹ J.B.R. Oonk,² A. Plaat,³ A. Danezi,² and T.W. Shimwell⁴

¹*Leiden Sterrewacht, Leiden, Zuid-Holland, Netherlands;*
apmechev@strw.leidenuniv.nl

²*SURFara, Amsterdam, Noord-Holland, Netherlands*

³*LIACS, Leiden, Zuid-Holland, Netherlands*

⁴*ASTRON, Dwingeloo, Drenthe, Netherlands*

Abstract. The LOFAR radio telescope is a low-frequency aperture synthesis radio telescope with headquarters in the Netherlands and stations across Europe. As a general purpose telescope, LOFAR produces petabytes of data each year serving a wide range of science cases. The data volumes produced are difficult or impossible to process on a single machine or even a small cluster at a scientific institute. We provide a layout for serving LOFAR processing to the astronomical community by providing access to LOFAR pipelines accelerated on a high throughput platform. We build this on our previous success with parallelizing the LOFAR Surveys pipeline and with creating automated LOFAR workflows on a distributed architecture. The LOFAR As A Service platform will serve the LOFAR Key Science Projects (KSPs), specifically the LOFAR Surveys KSP, which aims to provide science ready products to the scientific community. Additionally, this system will provide a robust method to re-process LOFAR data with a single click.

1. LOFAR

The LOFAR radio telescope (van Haarlem 2013) consists of more than 7000 antennae with a dense set of core stations near Exloo in the Netherlands, 14 remote Dutch stations and a further 13 international stations across Europe. LOFAR is an aperture synthesis array operating at low frequencies, between 10MHz and 250MHz. Producing a science quality LOFAR image requires large amounts of processing to remove image artifacts produced by the telescope and the ionosphere (van Weeren 2016). The data reduction required to create a scientific image typically requires 256 GB of RAM and 4 TB of disk space. These resources in practice equal one or more dedicated nodes on a modern computational cluster. Requiring dedicated hardware for each data set makes it difficult to perform large scale studies using the resources provided by a typical research institution (e.g., a university). These processing requirements can only be met by using a large scale shared high throughput platform. We deploy LOFAR processing on three such data centers located at the LOFAR Long-Term Archive (LTA) sites. The LTA data centres are at SURFara in Amsterdam, FZ-Jülich in Jülich and PSNC at Poznań with

the main deployment at SURFsara in the Netherlands¹. Through this work, large LOFAR projects like the LOFAR Two Meter Sky Survey (LoTSS) (Shimwell 2017) can integrate their processing pipelines with heterogeneous infrastructure provided natively at each of the LOFAR Archive locations.

1.1. LOFAR Surveys

The goal of the LOFAR Two Meter Sky Survey is to make broadband radio images of the Northern Sky at an unprecedented sensitivity. The survey is expected to produce more than 3000 observations, each of which is more than 16 TB. While the raw data is stored at three Long-Term Archive locations, moving and processing this data at an university institution is untenable. To ease this issue, we have developed a framework, GRID_LRT, to parallel process LOFAR data at the SURFsara gina High Throughput Cluster (Mechev et al. 2017). A diagram of the processing steps for one of the LOFAR pipelines is shown in Figure 1. Using this framework, it is easy to efficiently process LOFAR data with various scripts, making it possible to create data products serving multiple science cases.

2. Pipeline Parallelization and AGLOW

Using the GRID_LRT software², we distribute LOFAR processing at the SURFsara site. Because of the data-level parallelism of LOFAR data, some of the steps can be run in parallel, thus helping accelerate the processing. While the parallelization helps accelerate a single run of the pipeline, it is still necessary to facilitate scheduling and running the pipeline automatically. To do this, we built a software suite to interface LOFAR workflows with the grid middleware and LOFAR Long-Term Archive. This software, AGLOW, allows users to build high-level workflows and easily process them on an European grid e-infrastructure (Mechev et al. 2018).

3. LOFAR As A Service with AGLOW

The AGLOW software is built on Apache Airflow³ and has many options for triggering a workflow. Each AGLOW workflow can be scheduled automatically by the scheduler at a regular interval. Additionally, workflows can be triggered by another workflow or triggered manually by a user clicking the run button on the GUI or by the user launching the workflow through the CLI. Airflow also allows triggering of a workflow through a REST POST command, encoding parameters in the JSON payload.

Using this capability, we aim to integrate AGLOW with a lightweight web based front-end built on Django⁴. This front-end is responsible for authorizing the users, staging the requested data and passing the requested parameters to the respective AGLOW workflow. The front-end also presents the logs from the Grid jobs, AGLOW workflow

¹<https://www.surf.nl/en/about-surf/subsidiaries/surfsara/>

²https://github.com/apmechev/GRID_LRT

³<https://airflow.apache.org/>

⁴<https://www.djangoproject.com/>

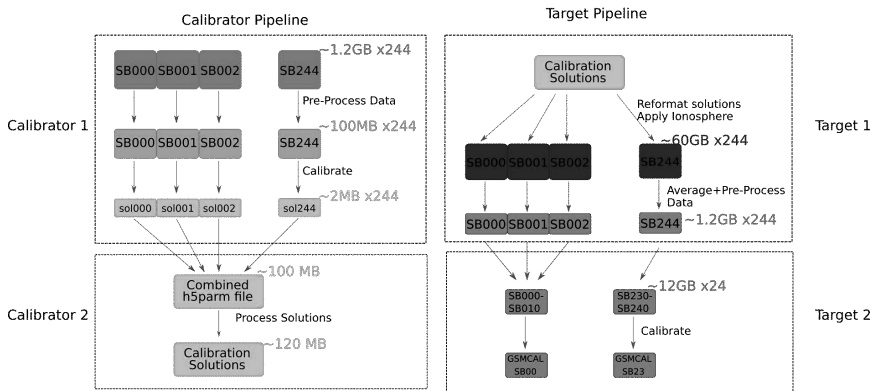


Figure 1. Parallelization for the LOFAR direction independent pipeline (de Gasperin et al. 2018).

and the front-end itself to the user. Additionally, these can be automatically sent to the user by email. Finally, the front-end can serve diagnostic plots to the users for each of their runs.

3.1. Adding and Modifying Pipelines

Each of the pipelines presented to the user contains a set of parameters. These parameters are sent to the AGLOW workflow and propagate to the relevant tasks. Ultimately, the worker nodes are fed the parameters and use them to process the requested data. Typical parameters are time and frequency averaging parameters which determine the time and frequency sampling rate of the final products. Other parameters include demixing sources, a string of sources that contaminate the observation and need to be explicitly removed. More complex pipelines can also request users to upload their own skymodels as text files, and use these skymodels to perform gain calibration on the requested data.

In order to add a new pipeline, the following steps need to be performed:

1. Define and test end-to-end pipeline execution at LTA sites
2. Decide on parameter types and values for pipeline steps
3. Implement pipeline in AGLOW including parameters (with default values)
4. Define JSON payload that will be sent by the front-end
5. Create a basic resource model describing processing time for different parameters (Typically data size, averaging parameters)
6. Add pipeline as an option to front-end

3.2. Performance Models

Processing LOFAR data can use significant computational resources. Presenting an easy to use interface to radio astronomers can lead to exhausting the requested resources at the institutions tasked with processing our data. Because of this, it is important to know roughly how many resources will be used by each job and prevent users from surpassing their allocated amount. As such, we aim to create simple processing models for each pipeline, and calibrate it as astronomers start using the service.

4. Conclusion and Future work

Creating a user-accessible portal for processing LOFAR data will be beneficial to the radio astronomy community, helping researchers efficiently process their data without having to dedicate computational resources at their universities, or spend time moving the raw data. This portal will include an authentication module, a workflow engine, and a performance model for each workflow. While the workflow engine exists and is currently in use by scientists, publishing such workflows to the broader community requires a model to predict the resources consumed by a single job based on the requested parameters. With such a model it will be possible to offer a managed service where the resources consumed by each project are measured and the job scheduling is optimized based on the remaining compute time and storage available on the shared infrastructure.

Acknowledgments. APM would like to acknowledge the support from the NWO/DOME/ IBM programme “Big Bang Big Data: Innovating ICT as a Driver For Astronomy”, project #628.002.001.

References

- de Gasperin, F., Dijkema, T. J., Drabent, A., Mevius, M., Rafferty, D., van Weeren, R., Brüggen, M., Callingham, J. R., Emig, K. L., Heald, G., Intema, H. T., Morabito, L. K., Offringa, A. R., Oonk, R., Orrù, E., Röttgering, H., Sabater, J., Shimwell, T., Shulevski, A., & Williams, W. 2018, ArXiv e-prints, arXiv:1811.07954. 1811.07954
- Mechev, A., Oonk, J. B. R., Danezi, A., Shimwell, T. W., Schrijvers, C., Intema, H., Plaat, A., & Rottgering, H. J. A. 2017, in Proceedings of the International Symposium on Grids and Clouds (ISGC) 2017, 2
- Mechev, A. P., Oonk, J. B. R., Shimwell, T., Plaat, A., Intema, H. T., & Röttgering, H. J. A. 2018, ArXiv e-prints, arXiv:1808.10735. 1808.10735
- Shimwell, T. W. e. a. 2017, A&A, 598, A104
- van Haarlem, M. P. e. a. 2013, A&A, 556, A2. 1305.3550
- van Weeren, R. J. e. a. 2016, The Astrophysical Journal Supplement Series, 223, 2

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Visualization in IRSA Services using Firefly

Emmanuel Joliet and Xiuqin Wu

*IPAC/Caltech, Pasadena, CA, USA; ejoliet@ipac.caltech.edu ,
xiuqin@ipac.caltech.edu*

Abstract. NASA/IPAC Infrared Science Archive (IRSA) curates the science products of NASA's infrared and submillimeter missions, including many large-area and all-sky surveys. IRSA offers access to digital archives through powerful query engines (including VO-compliant interfaces) and offers unique data analysis and visualization tools. IRSA exploits a re-useable architecture to deploy cost-effective archives, including 2MASS, Spitzer, WISE, Planck, and a large number of highly-used contributed data products from a diverse set of astrophysics projects.

Firefly is IPAC's Advanced Astronomy WEB UI Framework. It was open sourced in 2015, hosted at GitHub. Firefly is designed for building a web-based front end to access science archives with advanced data visualization capabilities. The visualization provide user with an integrated experience with brushing and linking capabilities among images, catalogs, and plots. Firefly has been used in many IPAC IRSA applications, in LSST Science Platform Portal, and in NED's newly released interface.

In this focus demo, we will show case many data access interfaces and services provided by IRSA based on Firefly. It will demonstrate the reusability of Firefly in query, data display, and its visualization capabilities, including the newly released features of HiPS images display, MOC overlay, and the interactions between all those visualization components.

1. Introduction

Archives are data driven end points which care mostly of curating, maintaining and making available reliably and as much as possible without interruption science data. Nowadays, network and internet is the main access and so UI and GUIs together with friendly APIs are the go-to tools for users

IRSA hosts more than 1PB of data from over 15 projects. It enable data extraction, exploration and visualization which required software development and maintenance (as well as hardware) for long-term persistence and availability. User experience (UX) learning curve is faster when User interfaces (UI) are consistent across websites. Best Software engineering (Scrum) rules and practices are put in place for such endeavor and growing challenges. Backend API are best companions to UI and are build behind for easy direct or internal access (i.e. VO protocols).

A simple view of archive access from internet clients is displayed below.

2. Challenges

Archive data usually can be retrieve and displayed as images, charts and tables. Interactivity and interconnectivity is a must in order to allow useful exploration and extraction. Always pursuing a "user friendly" access is key for enabling data exploration across different projects. Increasing data volume and complicated use cases are challenges that could be overcome by relying on services running closer to the data, with reusable and derived components across different projects and datasets.

Many projects in IPAC either use Firefly in applications or use the Firefly API for displays. They are WISE ¹, Herschel ², Finder Chart ³, IRSA Catalog Search ⁴, IRSA Viewer ⁵, and NED ⁶.

Using Firefly as a base provided a familiar look and feel to users, and allow all the projects to share the new features added to Firefly, like the most recent new feature of HiPS images display and MOC overlay.

Figure 1 "Firefly, IRSA catalog search" give two images. They are Firefly displaying a catalog entries on an image, color-color plot, and histogram, IRSA catalog search result display using Firefly API.

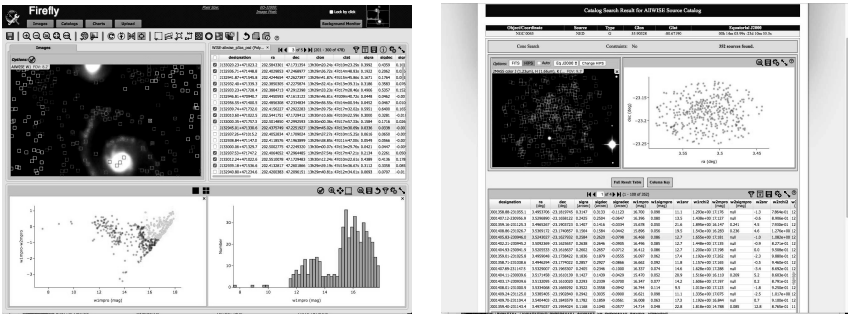


Figure 1. Firefly, IRSA catalog search

Figure 2 "WISE image service, Firefly" give two images. They are WISE image service to display multiple images for WISE, and 4 HiPS images.

3. Technical information

The IPAC Firefly is an open-source library to build UI core components based on ReactJS/Java Through collaborative development across IRSA, LSST and NED project, the

¹<https://irsa.ipac.caltech.edu/applications/wise/>

²<https://irsa.ipac.caltech.edu/applications/Herschel/>

³<https://irsa.ipac.caltech.edu/applications/finderchart/>

⁴<https://irsa.ipac.caltech.edu/applications/Gator/>

⁵<https://irsa.ipac.caltech.edu/irsaviewer/>

⁶<http://ned.ipac.caltech.edu>

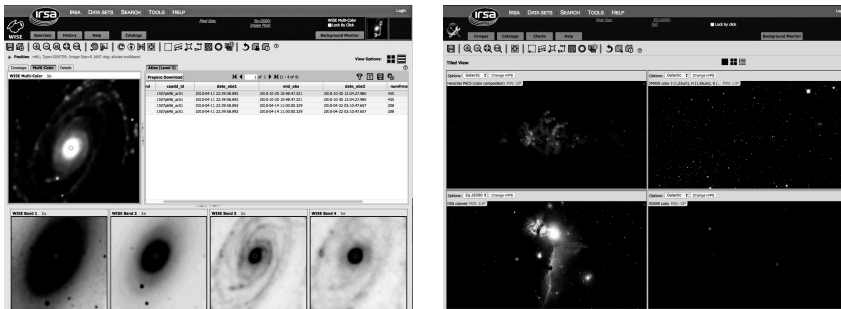


Figure 2. WISE image service, HiPS images

IPAC contributors are maintaining and adding features to the code. The code is hosted in a github repository⁷.

The library is used to build Web-based applications. It consists of a Server/client UI architecture sharing common library/stack (apache + tomcat), running HTML/JS client from ES6+(npm+redux+sagas+plotly+,etc.) and a Java layer on the server side, staging searches from DB/APIs/VO. We use Gradle/Jenkins for building and testing following the github pull request workflow with the help of staging builds using docker and kubernetes cluster.

The main widgets available are a FITS image viewer, tables and charts components. The main features are related to explore data using brushing and linking operations.

There are two kind of applications, on one hand there are science data tools such as Time Series tool⁸ for light-curve datasets and Finder Chart for cross-comparison of images from various surveys (+API). On the other hand there are project specific applications: i.e. WISE, Spitzer, Planck, Herschel, contributed products.

Recently we have added HIPs capabilities, a periodogram viewer, and more instrument footprints to be overlaid.

The library can be used in two ways. The components and their (re)actions are exposed via a higher level javascript API so HTML pages can import the library and make use of it directly. The Framework can be used to build and extend the existing React classes (low-level) via framework composition. The exposed properties are needed to control project specific requirements. Some of the widgets have particular options to be enabled or disabled depending on the project appearance context/decision. The objects inheritance help developers to maintain and add new features to the Firefly core.

4. Near future

Updates coming soon will include new available image datasets for searching IRSA archives, footprint overlay improved, MOC outline maps.

⁷<https://github.com/Caltech-IPAC/firefly>

⁸<https://irsa.ipac.caltech.edu/irsaviewer/timeseries>

We will be improving existing integration with other languages to enable science platform access to run code closer to data for mining and cross exploration with big-data. Python integration within notebooks/JupyterLab exists already and same UI widgets are exposed to allow multiple integration⁹. We will continue the effort to adopt modern web technology to enable richer features and take advantage of 3rd party libraries running in browsers.

5. Demo outline

A demo with step by step tutorial has been put together in github¹⁰. The tutorial should cover the following concepts:

- Brushing with histogram/charts: IV catalog search, with error bars, column expression (i.e. WISE)
- Gator -> Light-curve / Period finder with periodogram (WISE/PTF)
- HIPs demo , with URL (+ ivo://), ex: <https://irsa.ipac.caltech.edu/data/hips/list>
- Footprint overlay (JWST) - layers control
- API html integration: Atlas, Herschel or NED
- IRSA Finder Chart application
- Python integration
- In development: MOC, new Footprint tool

Acknowledgments. The Firefly software is based upon work supported in part by the National Science Foundation through Cooperative Support Agreement (CSA) Award No. AST-1227061 under Governing Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DEAC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.

The applications mentioned were supported by IRSA, the NASA/IPAC Infrared Science Archive. IRSA curates the science products of NASA's infrared and submillimeter missions, including many large-area and all-sky surveys, and NASA/IPAC's Extragalactic Database, NED.

⁹see Firefly Python client: https://github.com/Caltech-IPAC/firefly_client/blob/master/doc/index.rst

¹⁰<https://github.com/ejoliet/adass2018>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Image Processing in Python with Montage

John Good,¹ and G. Bruce Berriman²

¹*Caltech/IPAC-NExScI, Pasadena, CA 91125, USA; jcg@ipac.caltech.edu*

²*Caltech/IPAC-NExScI, Pasadena, CA 91125, USA*

Abstract. The Montage image mosaic engine¹ has found wide applicability in astronomy research, integration into processing environments, and is an exemplar application for the development of advanced cyber-infrastructure. It is written in C to provide performance and portability. Linking C/C++ libraries to the Python kernel at run time as binary extensions allows them to run under Python at compiled speeds and enables users to take advantage of all the functionality in Python. We have built Python binary extensions of the 59 ANSI-C modules that make up version 5 of the Montage toolkit. This has involved turning the code into a C library, with driver code fully separated to reproduce the calling sequence of the command-line tools; and then adding Python and C linkage code with the Cython library, which acts as a bridge between general C libraries and the Python interface.

We will demonstrate how to use these Python binary extensions to perform image processing, including reprojecting and resampling images, rectifying background emission to a common level, creation of image mosaics that preserve the calibration and astrometric fidelity of the input images, creating visualizations with an adaptive stretch algorithm, processing HEALPix images, and analyzing and managing image metadata.

The material presented here will be made freely available as a set of Jupyter notebooks posted on the Montage GitHub page.

1. Introduction

- Montage - image mosaic engine. Creates mosaics from input set of FITS images Written in ANSI-C. Portable. Components perform one task in the creation of a mosaic (list them). Plus utilities for managing and organizing files, managing FITS attributes, and analyzing image metadata. List them
- Wide applicability. Used in NEO detection, Instrument Performance, Observation planning for JWST, Citizen Science, Machine Learning.
- Created Python binary extensions of 59 modules in v5 of Montage. Gives users the power of Python at compiled speeds. This has involved turning the code into a C library, with driver code fully separated to reproduce the calling sequence of the command-line tools; and then adding Python and C linkage code with the Cython library, which acts as a bridge between general C libraries and the Python interface.

¹<http://montage.ipac.caltech.edu>; <https://github.com/Caltech-IPAC/Montage>

- The Python extensions have been released as v6 on Nov 12 2018 and tested on Python 3.6, Mac OS X.

Python binary extensions of existing Montage modules; no new functionality has been introduced. Click [here](#) to see a list (link to Jupyter notebook) of all supported modules. The Python extensions have been created by transforming the C code (<https://github.com/Caltech-IPAC/Montage>) into a library, with driver code fully separated to reproduce the calling sequence of the command-line tools; and then adding Python and C linkage code with the Cython library, which acts as a bridge between general C libraries and the Python interface. These binary extensions offer image processing at compiled speeds in the Python environment.

2. How To Install and Use It

There are two ways to install MontagePy, all of which include all supporting packages: there are no external dependencies. From PyPI, use the command "pip install MontagePy." Or download the .whl file and install with the command "pip install file.whl."

We have delivered a set of Jupyter notebooks (e.g., Figure 1) that give examples of how to use each component in Python, and compares usage in Python with that in C. The Jupyter notebooks are available for download at <https://github.com/Caltech-IPAC/MontageNotebooks> and they can be viewed without downloading at <http://montage.ipac.caltech.edu/MontageNotebooks>. A three color Sloan image made with mViewer is shown in Figure 2.

Acknowledgments. Montage is funded by the National Science Foundation under Grant Numbers ACI-1440620 and ACI-1642453., and was previously funded by the National Aeronautics and Space Administration's Earth Science Technology Office, Computation Technologies Project, under Cooperative Agreement Number NCC5-626 between NASA and the California Institute of Technology.

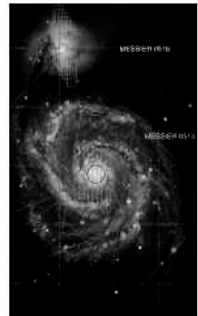
Building a Mosaic with Montage

Montage is a general toolkit for reprojecting and mosaicking astronomical images and generally you have to marshal the specific data you want to use carefully. But there are a few large-scale uniform surveys that cover a large enough portion of the sky to allow a simple location-based approach.

In this notebook we will choose a region of the sky and dataset to mosaic, retrieve the archive data, reproject and background-correct the images, and finally build an output mosaic. You are free to modify any of the mosaic parameters but beware that as you go larger all of the steps will take longer (possibly much longer). If you do this for three different wavelengths, you can put them together in a full-color composite using our Sky Visualization notebook, which produced the image on the right.

As with many notebooks, this was derived from a longer script by breaking the processing up into sequential steps. These steps (cells) have to be run one in sequence. Wait for each cell to finish (watch for the step number in the brackets on the left to stop showing an asterisk) before starting the execution of next cell or run them all as a set.

If you want to just see the code without all the explanation, check out this example.



Setup

The Montage Python package is a mixture of pure Python and Python binary extension code. It can be downloaded using `pip install MontagePy`

No other installations are necessary.

```
In [3]: # Startup. The Montage modules are pretty much self-contained
# but this script needs a few extra utilities.

import os
import sys
import shutil

from MontagePy.main import *
from MontagePy.archive import *

from IPython.display import Image

# These are the parameters defining the mosaic we want to make.

location = 'M 17'
size = 1.0
dataset = '2MASS J'
workdir = 'Messier017'
```

So not much to see so far. We've defined a location on the sky (which can be either an object name (e.g. "Messier 017") or coordinates. The coordinate parser is pretty flexible; "3h 29m 53s +47d 11m 43s" (defaults to the Equatorial J2000 system), "201.94201 47.45294 Equ B1950" and "104.85154 68.56078 Galactic" all work. We've also defined a size. In this case we are going to use this below to construct a simple North-up gnomonic projection square box on the sky; you are free to define any header you like as Montage supports all standard astronomical projections and coordinate systems.

Working Environment

Before we get to actually building the mosaic, we need to set up our working environment. Given the volume of data possible, the Montage processing is file based and we need to set up some subdirectories to hold bits of it. This will all be under an instance-specific directory specified above ("workdir"). It is best not to use directory names with embedded spaces.

```
In [4]: # We create and move into subdirectories in this notebook
# but we want to come back to the original startup directory
# whenever we restart the processing.
```

Figure 1. Section of Jupyter Notebook for Building a Mosaic of M17


```
In [16]: try:
          os.makedirs('work/SDSS')
        except:
            pass

        rtn = mViewer(imgjson, 'work/SDSS/SDSS.png', mode=1)

        print(rtn)

{'status': '0', 'type': 'b'color', 'nx': 1200, 'ny': 1200, 'grayminval': 0.0, 'grayminpercent': 0.0, 'grayminsigma': 0.0, 'graymaxval': 0.0, 'graymaxpercent': 0.0, 'graymaxsigma': 0.0, 'blueminval': 1660.948886289596, 'blueminpercent': 45.51304816901208, 'blueminsigma': -0.100000000000002897, 'bluemaxval': 18833.51097929482, 'bluemaxpercent': 100.0, 'bluemaxsigma': 7293.270993055378, 'greenminval': 612.1976698225872, 'greenminpercent': 43.049896485589686, 'greenminsigma': -0.10000000000000262, 'greenmaxval': 26276.085778495264, 'greenmaxpercent': 100.0, 'greenmaxsigma': 14790.776567073211, 'redminval': 896.9336930051037, 'redminpercent': 42.62543763181061, 'redminsigma': -0.10000000000000908, 'redmaxval': 20567.63987893109, 'redmaxpercent': 100.0, 'redmaxsigma': 6285.599891788824, 'graydatamin': 0.0, 'graydatamax': 0.0, 'bdatamin': 1650.1333317872777, 'bdatamax': 18833.51097929482, 'gdatamin': 608.5073366952034, 'gdatamax': 26276.085778495264, 'rdatamin': 890.1481250441316, 'rdatamax': 20567.63987893109, 'flipX': 0, 'flipY': 1, 'colortable': 0, 'bunit': 'b'}
```

The return contains details of the individual image stretching, image size, and so on.

Here is the output image:

```
In [17]: from IPython.display import Image
         Image(filename='work/SDSS/SDSS.png')
```

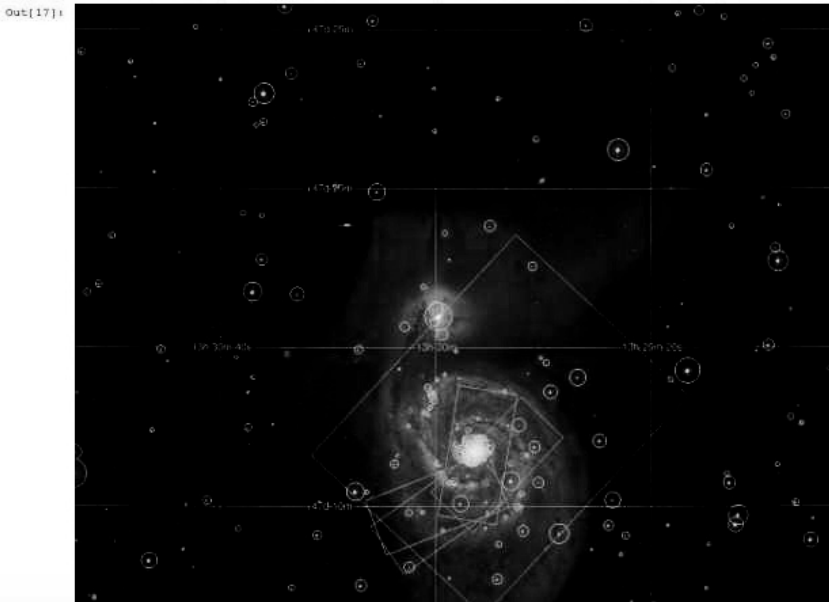


Figure 2. Three color Sloan Image of M51 Created With mViewer

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Workflows using Pegasus: Enabling Dark Energy Survey Pipelines

Karan Vahi,¹ Michael H Wang,² Chihway Chang,³ Scott Dodelson,⁴ Mats Rynge,¹ and Ewa Deelman¹

¹*USC Information Sciences Institute, Marina Del Rey, California, USA;*
vahi@isi.edu, rynge@isi.edu, deelman@isi.edu

²*Fermi National Laboratory, Batavia, Illinois, USA; mwang@fnal.gov*

³*Kavli Institute of Cosmological Physics - University of Chicago, Chicago, Illinois, USA; chihway@kicp.uchicago.edu*

⁴*Carnegie Mellon University, USA; sdodelso@andrew.cmu.edu*

Abstract. Workflows are a key technology for enabling complex scientific applications. They capture the inter-dependencies between processing steps in data analysis and simulation pipelines, as well as the mechanisms to execute those steps reliably and efficiently in a distributed computing environment. They also enable scientists to capture complex processes to promote method sharing and reuse and provide provenance information necessary for the verification of scientific results and scientific reproducibility. We describe a weak-lensing pipeline that is modeled as a Pegasus workflow with pipeline codes available as a Singularity container. This has enabled us to make this analysis widely available and easily replicable to the astronomy community. Using Pegasus, we have executed various steps of pipelines on different compute sites with varying infrastructures, with Pegasus seamlessly managing the data across the various compute clusters in a transparent manner.

1. Introduction

One of the most exciting and challenging areas of modern cosmology is weak gravitational lensing: the phenomenon of small distortions in the shapes of background galaxies as the light they emit traverses the lumpy universe. By measuring the shapes of galaxies in a given region, cosmologists can infer the total mass along the line of sight. Carried out over large parts of the sky, this inference can shed light on the most profound mysteries in the universe, such as the nature of dark matter and dark energy. The analyses are so complex that the process of making the measurements on petabytes of imaging data, processing the measurement outputs, applying algorithms that measure shapes of billions of galaxies, and then using that information to learn about the universe takes years of effort by large collaborations to carry out. In order to make this analysis widely available and easily replicable to the astronomy community, we decided to model this pipeline as a Pegasus workflow.

Pegasus WMS (Deelman et al. 2015) is a workflow management system that can manage large-scale pipelines across desktops, campus clusters, grids and clouds. In Pegasus WMS, pipelines are represented in an abstract form that is independent of the

resources available to run it and the location of data and executables. It compiles these abstract workflows to executable workflows represented as HTCondor DAG's that can be deployed onto distributed and high-performance computing resources such as DOE LCFs like NERSC, XSEDE, local clusters, and clouds. The executable form is an advanced DAG which is executed by HTCondor DAGMan. Pegasus is used in a number of scientific domains doing production grade science. In 2016 the LIGO gravitational wave experiment used Pegasus for its PyCBC pipeline (Usman et al. 2016) to analyze instrumental data and confirm the first ever detection of a gravitational wave. The Southern California Earthquake Center (SCEC) based at USC, uses a Pegasus managed workflow infrastructure called CyberShake to generate hazard maps for the Southern California region. In March 2017, SCEC conducted a CyberShake study (Callaghan et al. 2017) on DOE systems ORNL Titan and NCSA BlueWaters to generate the latest maps for the Southern California region. Overall, the study required 450,000 node-hours of computation across the two systems. Pegasus is also being used in astronomy, bioinformatics, civil engineering, climate modeling, earthquake science, molecular dynamics and other complex analyses.

2. Benefits of Pegasus Approach for Compute Pipelines

HTCondor DAGMan (Thain et al. 2005) is a common foundation for many astronomy pipelines. When using HTCondor DAGMan directly, astronomy projects commonly find themselves developing pipelines for a particular execution environment and data storage solution, and therefore have to spend valuable development time to continuously adjust the pipeline for new infrastructures or changes to the existing infrastructure. Pegasus WMS solves this problem by providing an abstraction layer on top of HTCondor DAGMan.

Using Pegasus instead of writing out HTCondor DAGs directly, enables us to execute the pipeline on a distributed grid infrastructure, where Pegasus takes care of the data management of the pipeline. During the mapping step, Pegasus WMS tries to look up the LFNs to obtain a list of physical file names (PFN) which are URLs to locations of where the file can be found. For input files, the system determines the appropriate replica to use, and which data transfer mechanism to use. Data transfer tasks are added to the pipeline accordingly. If during the lookups, Pegasus WMS finds that a subset of the pipeline outputs already exists, the pipeline will be automatically pruned to not recompute the existing outputs. This data reuse feature is commonly used for projects with overlapping datasets and pipelines. Required data transfers are automatically added to the pipeline and optimized for performance. Pegasus WMS imports compute environment descriptions provided by the scientist, and URLs for the data to schedule data transfers, including credential management. During the mapping step, it is determined when intermediate data files are no longer required. Clean up tasks are added, with the overall result being a minimized data footprint during execution. Data registration is the feature that adds information about generated outputs to an information catalog for future data discovery and potential data reuse.

In order to keep the science code dependencies portable and easily deployable, Pegasus allows scientists to package their science code and dependencies into a Docker or a Singularity container. The use of application containers ensures the scientific code is executed in a homogeneous environment tailored for application, even when executing on nodes with widely varying architecture, operation systems and system libraries. The

container images themselves then managed and deployed on the fly on the nodes where the jobs execute automatically by Pegasus.

3. Weak Lensing Pipeline

The pipeline described here is an example of a typical gravitational weak lensing analysis. It uses publicly available Science Verification catalogs(team 2018) of the Dark Energy Survey (DES-SV).

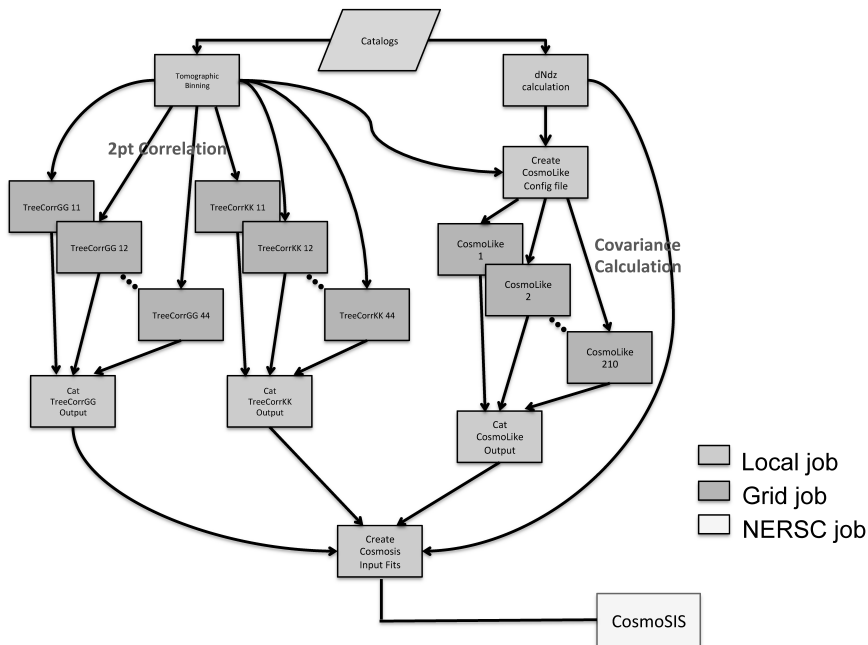


Figure 1. Weak Lensing Pipeline deployed at FNAL

The very first two steps of the pipeline are `doTomoBinning` and `calDndz`, which directly read from the DES-SV input catalogs. `doTomoBinning` selects and sorts the objects in the input shape catalog into *nbtomo* tomographic bins, writing out smaller fits files for each bin. `calDndz` sums up the PDFs for each galaxy in the input catalog to calculate the full redshift distribution. After `doTomoBinning` completes, $N_c = nbtomo(nbtomo + 1)/2$ *TreeCorr*(Jarvis 2018) jobs are launched in parallel to calculate the two-point shear correlation functions using the fits files produced by `doTomoBinning` as input.

After both `doTomoBinning` and `calDndz` are done, *CosmoLike* is launched to calculate the analytic covariance associated with the data vector. It uses the redshift distributions from `calDndz` and the effective number density, total shape noise, and survey area from `doTomoBinning` as input. A total of $N_c(2N_c + 1)$ *CosmoLike*(Krause & Eifler 2017) jobs are launched in parallel to calculate all the sub-matrices of the full covariance matrix.

After TreeCorr and CosmoLike complete, their results are each concatenated into two separate files which, together with the redshift distributions from calDndz, serve as input for mkCosmosisFits. This last step produces an output fits file that has all the relevant information arranged in a format supported by the CosmoSIS(Zuntz et al. 2015) framework used to extract the cosmological parameters.

We have used the pipeline described above to carry out a unified analysis(Chang et al. 2019) of four recent cosmological datasets (DLS, CFHTLenS, DES-SV, KiDS-450). The modular and flexible framework of the pipeline structure allows us to perform the analysis in a systematic way that uses the computational resources effectively and is easy to debug. Pegasus enabled us to execute various steps of pipelines illustrated in Figure 1 on different compute sites (HTCondor pool Grid at FNAL, local workflow server and NERSC), with Pegasus seamlessly managing the data across the various compute clusters in a transparent manner. As part of the unified analysis, about 3,041 HTCondor jobs were executed using about a year's worth of computing time. A demonstrative version of this pipeline is available on GitHub (Wang & Vahi 2018) for users to download and perform similar analysis. For the focus demonstration, we used a small HTCondor pool at USC/ISI, configured with 128 slots. The compute nodes had Singularity pre-installed.

Acknowledgments. Pegasus is funded by the National Science Foundation(NSF) under the OAC SI2-SSI #grant 1664162. Previously, NSF has funded Pegasus under OCI SDCl program grant #0722019 and OCI SI2-SSI program grant #1148515.

References

- Callaghan, S., Juve, G., Vahi, K., Maechling, P. J., Jordan, T. H., & Deelman, E. 2017, in 12th Workshop on Workflows in Support of Large-Scale Science (WORKS'17)
- Chang, C., Wang, M., & Dodelson, S. e. a. 2019, Monthly Notices of the Royal Astronomical Society, 482, 3696. /oup/backfile/content_public/journal/mnras/482/3/10.1093_mnras_sty2902/1/sty2902.pdf, URL <http://dx.doi.org/10.1093/mnras/sty2902>
- Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., Mayani, R., Chen, W., Ferreira da Silva, R., Livny, M., & Wenger, K. 2015, Future Generation Computer Systems, 46, 17. URL <http://pegasus.isi.edu/publications/2014/2014-fgcs-deelman.pdf>
- Jarvis, M. 2018, Code for efficiently computing 2-point and 3-point correlation functions., <https://github.com/rmjarvis/TreeCorr>
- Krause, E., & Eifler, T. 2017, Monthly Notices of the Royal Astronomical Society, 470, 2100. /oup/backfile/content_public/journal/mnras/470/2/10.1093_mnras_stx1261/1/stx1261.pdf, URL <http://dx.doi.org/10.1093/mnras/stx1261>
- team, D. 2018, Public DES-SV data, <https://des.ncsa.illinois.edu/releases/sva1>
- Thain, D., Tannenbaum, T., & Livny, M. 2005, Concurr. Comput. : Pract. Exper., 17, 323. URL <http://dx.doi.org/10.1002/cpe.v17:2/4>
- Usman, S. A., Nitz, A. H., Harry, I. W., Biwer, C. M., & et al, D. A. B. 2016, Classical and Quantum Gravity, 33, 215004. URL <http://stacks.iop.org/0264-9381/33/i=21/a=215004>
- Wang, M. H., & Vahi, K. 2018, Executing Weak Lensing Pipelines using Pegasus, <https://github.com/pegasus-isi/pegasus-wlpipe>
- Zuntz, J., Paterno, M., Jennings, E., & et al, D. R. 2015, Astronomy and Computing, 12, 45 . URL <http://www.sciencedirect.com/science/article/pii/S2213133715000591>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

AAS Journals: Software and Data

F.X. Timmes¹ and August Muench¹

¹*American Astronomical Society, Washington, D.C., USA; frank.timmes@aaas.org*

Abstract. Software and data are an integral enabler of observation, experiment, theory, and computation and a primary modality for realizing discoveries and innovations. The ongoing explosion of activity in multi-messenger astronomy powers theoretical and computational developments, in particular the evolution of the community-driven software and data instruments. The AAS Journals welcomes and supports publication of software and data instrument papers.

1. AAS Journals Software and Data Papers

The American Astronomical Society Journals (*The Astronomical Journal*, *The Astrophysical Journal*, *The Astrophysical Journal Letters*, *The Astrophysical Journal Supplements*, and *Research Notes*) welcomes and supports articles which describe the design and function of software or data of relevance to research in astronomy and astrophysics. To support this initiative the AAS Journals established a new corridor (Vishniac & Lintott 2016) entitled Instrumentation, Software, Laboratory Astrophysics, And Data that approximately corresponding to IAU Division B: Instrumentation, Software, Laboratory Astrophysics, and Data. Example publications include the recent Astrophysical Journal Supplement Special Issue, Data: Insights and Challenges in a Time of Abundance (Timmes & Golub 2018), where 23 articles explore a range of topics associated with data-driven discovery and analysis that span the AAS journal topical corridors.

2. Guidelines for Software and Data Articles

Submitted articles should contain a description of the software or data instrument, its novel features and its intended use. Articles need not include research results produced using the software or data, although including example applications can be useful.

If a piece of novel software or data science is important enough to published research then it is likely appropriate to describe the software or data science in such an article. We recommend that authors release source code described in a submitted article under an appropriate open source license (see <http://opensource.org/faq#osd> or <http://choosealicense.com/>) and archive the published version of their source code using a service such as Zenodo (<https://zenodo.org/>) or FigShare (<http://figshare.com/>) which will provide a unique digital object identifier (DOI) and ensure that the code is accessible in the long term. However, any articles which provide a clear statement on how to access the code - for example, by contacting the author - are acceptable. Work-

flows for publishing code with a DOI include the article Making your Code Citable” (<https://guides.github.com/activities/citable-code/>) from GitHub and Zenodo.

3. Guidelines for Citation of Software and Data

Software and data can be cited in two ways: 1) Citing the article describing the software. For example, “The Astropy Project: Building an Open-Science Project and Status of the v2.0 Core Package”, The Astropy Collaboration et al, 2018, *AJ*, 156, 123, doi:10.3847/1538-3881/aabc4f, and 2) Citing a DOI for the software obtained, for example, via Zenodo or FigShare. For example, Ginsburg et al. 2018, *Astropy/Astroquery*, v0.3.8, Zenodo, doi:10.5281/zenodo.1234036, as developed on GitHub.

Ideally, both forms of citation should be included. The first extends credit to the authors for their publication and tells the reader where to learn about the software or data instrument. The second gives the reader access to the exact version of the software or data used in the project. These forms of citation are intended to allow authors to properly reference their use of software or data; alongside these formal references, they may also want to include links to code repositories such as GitHub, or code indices such as the Astrophysics Source Code Library (ASCL, <http://ascl.net/>).

Authors should endeavor to discover or identify the software developers’ preferred form of citation and use this in their papers and reference lists. The developers’ preferred citation is often found on a code repository landing page or in the documentation and could point to any of a software paper, a code DOI, or an indexed entry in the ASCL. Thus, a very good source of preferred citations is the ASCL. The ASCL actively curates a preferred citation field for the codes in the index by parsing and collating this information from the code repositories, making discovery of such references easier.

Authors may also include a section below the acknowledgments listing scientific software packages used as part of the work presented in the manuscript. This should be done via the new `\software` AASTeX 6 macro. The content of the command should take the form of a list of software name and citation in parentheses, for example:

```
\software{Astropy\citep{https://doi.org/10.3847/1538-3881/aabc4f},
          Matplotlib\citep{https://doi.org/10.1109/MCSE.2007.55}}
```

This is analogous to acknowledging a major facility or hardware instrument and is done for the same reason, to give credit to a project which is useful to the community. A code listed in `\software` need only be research software used in the analysis shown in the paper. It need not be previously mentioned in the main text.

4. Partnership with the Journal of Open Source Software

The AAS Journals has partnered with the Journal of Open Source Software (JOSS, <https://joss.theoj.org>), who can provide a software review for consideration by the Scientific Editor (<https://journals.aas.org/editorial/>) of the submitted manuscript.

References

- Timmes, F., & Golub, L. 2018, *ApJS*, 236, 1
 Vishniac, E. T., & Lintott, C. 2016, *AJ*, 151, 21

Session XVI

Birds of a Feather

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Open Source/Development Software Projects and Large Organizations/Missions: Recommendations and Challenges

Erik Tollerud and Steve Crawford

Space Telescope Science Institute, Baltimore, MD, USA; etollerud@stsci.edu

Abstract. Independent open-developed projects have been growing rapidly in astronomy and related domains - e.g., projects like Astropy, Sunpy, or Scipy. Large astronomy organizations or missions regularly make use of the software products of these projects, and at least in some cases, contribute back. This has the obvious benefit of being able to do more for the scientific community with less effort per mission. However, there are some challenges in this process: inflexible or aggressive deadlines by funded organizations may conflict with the timelines of open source projects, science vs engineering cultural conflicts may make contribution more difficult, mismatch between the needs of the general community and a specific mission. This BoF session was aimed at discussing exactly these tensions, some recommendations for how to resolve some of these tensions, and “sales pitches” for helping missions/institutions understand why it might be worth their time to contribute to open source projects.

Open-Developed Software (ODS)¹ and Open Source Software (OSS) projects have become common in the astronomy software community, especially in the Python ecosystem (e.g. Astropy Collaboration et al. 2018; SunPy Community et al. 2015; Jones et al. 2011). This means such projects must interface with institutes or missions that produce science tools that interact with or depend on the code these projects produce (e.g. the Large Synoptic Survey Telescope, James Webb Space Telescope, etc.). The Birds of a Feather (BoF) session reported on here was aimed at discussing the tensions and possible solutions to the tensions between these projects and missions. This proceeding reports on both the results itself and a survey provided to the BoF participants (with a response rate of $n = 13$, spanning a range of scales of mission or institute).

1. Adoption of Open Source Software by Missions

The survey results demonstrated a large uptake of OSS/ODS among missions. With only one exception, the participants reported their missions depend heavily on open source software (average ranking 4.5 where 5 was “agree” and 1 was “disagree”). Similarly, most report that the software they *produce* is also generally open source (average=4.1). This demonstrated that the missions and institutes already understand the value of such software, and in general a sales pitch on the value of such software is more focused on the details of *how* to interface with the projects rather than *why* such

¹Open-developed software is software where code contributions are open to all and most (or all) decisions about the code are made publicly via open discussion with the user/developer community.

interaction is needed. That said, a few points were highlighted in discussion where missions particularly value OSS:

- Provides built-in user feedback before missions are actually completed (because the software is already in use by the user community).
- Shares development resources with the wider science community, allowing missions to do more with less software budget.
- Simplifies training of the scientific user community in how to use the tools, as the training burden is shared across institutions.
- Improves recruiting for institutions by way of an extended network of developers with knowledge of relevant software.

2. Challenges/Recommendations of ODS for Missions

By contrast, the survey revealed a smaller (but not insignificant: average=3.4) fraction have embraced an ODS ethic (i.e., where the community can contribute to the mission's software). Similarly, a slightly smaller set (average=3.1) contribute back regularly to the community projects they use. This was reflected in a focus in the session's discussion on the challenges (and possible resolutions) of such engagement.

The origin of this was not focused in a particular area. In some cases the concerns had no obvious resolution, and therefore might simply be areas where ODS/OSS was not as well suited. For example, software developed by a mission was simply too specific to be open-developed (e.g., specific to a particular piece of instrumentation hardware). For others the concern was more driven by science-based concerns that the mission itself could not really address - e.g., a scientific community with a fear of being "scooped" that therefore did not want any code to be available outside the team. Because the session was seeking solutions, much of the discussion was focused in more tractable areas.

One of the significant concerns raised for using externally-developed OSS was the conflict between what the community desires and the immediate needs of missions. For example, the community for an OSS project might start moving towards a framework that the mission or institute judges to be difficult to support long-term. While rare, the specific scenarios highlighted the fact that in general OSS projects have a vested interest in the missions helping to support them. As a result they tend to be willing to accept quite significant changes if it is needed by a mission so as to build a long term relationship. This is particularly true for ODS projects, where a mission having a stake often gives them *direct* influence over how decisions are made in the project. An additional solution suggested (and generally agreed on) was to make sure it is understood from the outset that development in one direction does not preclude the other. That is, embracing a particular OSS solution does not mean the mission is bound to accept other OSS by the same community. If that is recognized, there is little-to-no downside as the mission can at any time move in an independent direction while still gaining the benefit of the work up until that point. While not as desirable for sharing effort across the community, it was judged to be sufficient to help missions see how this at least is not an impediment to *using* OSS/ODS.

Another concern raised for both using OSS and adopting ODS by missions was that of security risks. While the discussion was focused on the assumption of a bad actor, there was agreement that these suggestions applied to the more common scenario of *accidental* security breaches. While, again, there were relatively few specific examples of this happening, the recommendations were quite clear. Specifically, that following ODS principles often result in *improvement* over closed practices. This is because most ODS approaches require both code review and continuous integration (CI) to ensure their stability. This is a higher standard than many present-day missions and institutes, and therefore adopting these practices can serve to *strengthen* security rather than lower it. Therefore the suggestion reached was that institutes interested in adopting OSS or ODS should ensure they follow up-to-date practices of code review and CI as a part of their transition to such processes.

An additional concern discussed at length was the question of how missions that wish to follow more open practices should deal with the problem that they may need to say “no” to some contributions. That is, a community contribution may not be up to the standards or expectations of the mission. The mission’s representative needs a way to reject such a contribution without breaking the social contract of an ODS process. While there was no hard-and-fast answer to this, several solutions were suggested. One possibility was to accept part of the contribution, at some added effort by the mission to split out the contribution into a “good” and “bad” portion. Another option suggested was to be sure to try to convince the contributor of the problem, but if they don’t accept it, at some point simply state that the contribution cannot be accepted because it doesn’t meet the mission’s requirements. Regardless of the solution to any particular case, though, there was consensus in the discussion that a key element is setting clear expectations. That is, a mission that wishes to adopt or take part in ODS processes should set or know the “ground-rules” with, if at all possible, a written-out set of contributor guidelines². Some examples of such guidelines were discussed, but the general understanding was that the rules are often specific to the needs of the community or mission. It was generally agreed that the existence of and enforcement of such guidelines address most of the possible conflict from contributions not going as expected. Hence this came out as one of the strongest recommendations: if you want contributors, have contributor guidelines.

All that said, one of the largest concerns highlighted by the participants was not an unwillingness to contribute, but rather a lack of time. That is, missions/institutes focus on their requirements, and even when they are fully behind ODS/OSS projects they must focus on mission priorities. This demonstrated the importance of having a clearer idea of what level of investment of resources should be recommended for institutes to “buy into” an external OSS project, an idea that will hopefully be developed further in BoFs at future ADASSes.

While open development has not been widely adopted by missions yet, it offers an opportunity for greater collaboration and better science. The collaboration made possible through open development provides greater transparency and review of the software being developed and used. It thereby lowers development costs and leverages

²E.g., the Astropy contributor guidelines: <https://github.com/astropy/astropy/blob/master/CONTRIBUTING.md>, the JWST pipeline contributing guide: <https://github.com/spacetelescope/jwst/blob/master/CONTRIBUTING.md>, or the LSST documentation guidelines: https://github.com/lsst/pipelines_lsst_io/blob/master/.github/CONTRIBUTING.rst

the community to generate the best possible solutions for different problems. In the end that advantage was recognized by many participants and its future seems bright.

References

- Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., Ardelean, C., Babej, T., Bach, Y. P., Bachetti, M., Bakanov, A. V., Bamford, S. P., Barentsen, G., Barmby, P., Baumbach, A., Berry, K. L., Biscani, F., Boquien, M., Bostroem, K. A., Bouma, L. G., Brammer, G. B., Bray, E. M., Breytenbach, H., Buddelmeijer, H., Burke, D. J., Calderone, G., Cano Rodríguez, J. L., Cara, M., Cardoso, J. V. M., Cheedella, S., Copin, Y., Corrales, L., Crichton, D., D'Avella, D., Deil, C., Depagne, É., Dietrich, J. P., Donath, A., Droettboom, M., Earl, N., Erben, T., Fabbro, S., Ferreira, L. A., Finethy, T., Fox, R. T., Garrison, L. H., Gibbons, S. L. J., Goldstein, D. A., Gommers, R., Greco, J. P., Greenfield, P., Groener, A. M., Grollier, F., Hagen, A., Hirst, P., Homeier, D., Horton, A. J., Hosseinzadeh, G., Hu, L., Hunkeler, J. S., Ivezić, Ž., Jain, A., Jenness, T., Kanarek, G., Kendrew, S., Kern, N. S., Kerzendorf, W. E., Khvalko, A., King, J., Kirkby, D., Kulkarni, A. M., Kumar, A., Lee, A., Lenz, D., Littlefair, S. P., Ma, Z., Macleod, D. M., Mastroiello, M., McCully, C., Montagnac, S., Morris, B. M., Mueller, M., Mumford, S. J., Muna, D., Murphy, N. A., Nelson, S., Nguyen, G. H., Ninan, J. P., Nöthe, M., Ogaz, S., Oh, S., Parejko, J. K., Parley, N., Pascual, S., Patil, R., Patil, A. A., Plunkett, A. L., Prochaska, J. X., Rastogi, T., Reddy Janga, V., Sabater, J., Sakurikar, P., Seifert, M., Sherbert, L. E., Sherwood-Taylor, H., Shih, A. Y., Sick, J., Silbiger, M. T., Singanamalla, S., Singer, L. P., Sladen, P. H., Sooley, K. A., Sornarajah, S., Streicher, O., Teuben, P., Thomas, S. W., Tremblay, G. R., Turner, J. E. H., Terrón, V., van Kerkwijk, M. H., de la Vega, A., Watkins, L. L., Weaver, B. A., Whitmore, J. B., Woillez, J., Zabalza, V., & Astropy Contributors 2018, *AJ*, 156, 123. 1801.02634
- Jones, E., Oliphant, T., Peterson, P., et al. 2001–, *SciPy: Open source scientific tools for Python*. [Online; accessed <today>], URL <http://www.scipy.org/>
- SunPy Community, T., Mumford, S. J., Christie, S., Pérez-Suárez, D., Ireland, J., Shih, A. Y., Inglis, A. R., Liedtke, S., Hewett, R. J., Mayer, F., Hughitt, K., Freij, N., Meszaros, T., Bennett, S. M., Malocha, M., Evans, J., Agrawal, A., Leonard, A. J., Robitaille, T. P., Mampaey, B., Iván Campos-Rozo, J., & Kirk, M. S. 2015, *Computational Science and Discovery*, 8, 014009. 1505.02563



Eric Tollerud and Steve Crawford chairing the “OSS” BoF (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

Data Formats BoF

Jessica Mink,¹ Rosa Diaz², Keith Shortridge³, and Tim Jenness⁴

¹*Smithsonian Astrophysical Observatory, Cambridge, MA, USA;*
jmink@cfa.harvard.edu

²*Space Telescope Science Institute, Baltimore, MD, USA; rdiaz@stsci.edu*

³*K&V, Sydney, New South Wales, Australia; keithshortridge@gmail.com*

⁴*Large Synoptic Survey Telescope, Tucson, AZ, USA; tjenness@lsst.org*

Abstract. Nothing has really changed in the FITS format over the past year, so we concentrated on what would improve FITS as an exchange format to be used with the more complicated data structures which are being used with new telescopes such as JWST and LSST. By what is turning out to be a simple change of allowing header keywords to be longer than 8 characters, both data producers and data users will be happier with FITS. We also worked toward standardization of parameters where possible in the new structured data formats as they mature, possibly under the auspices of the new IAU Working Group on Astronomical Data Representation.

1. Improving FITS

In 2017's annual Birds of a Feather discussion about data formats (Mink 2019a), we discussed a proposed change to the FITS standard to accommodate keywords longer than 8 characters. The consensus of those in attendance was that changes to the FITS standard, and importantly, software implementing the FITS standard, should be minimal. It was felt that the ESO HIERARCH format (Wicenec et al. 2009b), which already allows long keywords in a standardized format, is already registered as a FITS convention (Wicenec et al. 2009a), and is supported by multiple software packages, such as CFITSIO (Pence 2017), should simply be added to the standard. A keyword line in that format appears in a FITS header as follows

```
HIERARCH token_1 token_2 ... token_n = value comment
```

where token_n can be of arbitrary length, as long as the total length of the line remains less than 80 characters. Since the keyword portion of the line often takes up quite a few characters, a string value can easily flow to another line, requiring the CONTINUE keyword to be used to accommodate <keyword>=<value>/<comment> lines over 80 characters. Jessica Mink discovered that implementing HIERARCH keywords effectively required the same parsing of header lines needed to implement keywords of arbitrary reasonable lengths., so one can simply do this:

```
LONGKEYWORDNAME = value / comment
```

Her work is described in this year's WCSTools poster on WCSTools (Mink 2019b).

Bill Pence remarked that long keywords are in CFITSIO already but a switch needs to be flipped to turn them on. If you allow long keywords, allowing spaces in them can be done if there is always '=' or '/', and we have to treat HIERARCH as a special case. He added that if a header reader does not find "=" in the 9th or 10th column, it should realize that there is no valid keyword on that line and take it as a comment. He also noted that this has not always been implemented, so old FITS readers could break. Blank lines should assumed to be comments, too.

There was a consensus that we should add long keywords to the FITS standard, sticking to UPPERCASE characters, and possibly allowing additional special characters, following the rule that degrees of freedom which are not needed should not be implemented.

2. Structured Data Formats

Our ongoing discussion about standardizing alternatives to the FITS format which better met the needs of large projects continued. The people in attendance who are working on new instruments are using data structures more complicated than FITS, though FITS files are often used as part of the larger structure. At this point, each large project seems to be moving in its own direction. STScI is using ASDF (Greenfield et al. 2015) (YAML + Binary and also JSON schema) to store the WCS information. Rosa Diaz noted that you cannot represent their WCS in FITS even if you allow for variable length keywords. LSST is considering the use of HDF5 and is using a data abstraction layer that hides the file format from the pipeline code (Jenness et al. 2019). For data processing, HDF5 and FITS could then both be options. ALMA data is stored in ALMA Science Data Model (ASDM) (Morales 2012), a directory hierarchy that includes binary and XML files. Anne Raugh reported that the Planetary Data System (PDS) defines special formats for archival purposes only. Data structures can be in FITS (Marmo et al. 2018), but there is a general lack of formalized metadata.

A particle physicist came to hear about a recommended format to use. Their data seems to be variable in format (number of pixels, number of cameras) and is time series information. For a dataset, they join the data. FITS does not work because it is not flexible to use. HDF5 is not perfect, either. Their data is close to that of X ray telescopes but more complex.

We now have two libraries that can interpret complex WCS using chains of mappings in serial and in parallel; the gWCS library from STScI (Dencheva 2019) and Starlink AST (Berry et al. 2016). Now that the second STC model has been developed within the IVOA and includes WCS transformations, we are in a position where we should consider adding the ability for gWCS and AST to exchange WCS transformations using the STC-2 data model, serialized in a well-documented intermediary format.

Demonstrating the efficacy of this interoperability will encourage other tools such as Aladin and DS9, the latter of which already uses AST, to be updated to recognize this serialization format. STScI, LSST (users of AST), and IVOA are actively discussing the possibility of using STC-2 as an interchange format.

We need to find a serialization format that works for all. It was strongly suggested that the IVOA WCS model become a standard between projects; however, it is currently tied to data format or VOTable using XML schema. Could there be an independent format to serialize the WCS info, such as an YAML or JSON string format that everybody

can read with IVOA model serialization of WCS? The LSST, STScI and STC-2 teams will be talking about that and experimenting.

3. IAU Data Representation Working Group

Lucio Chiappetti, who chaired this group two years ago, thinks that we should try to resurrect both the International Astronomical Union Data Representation Working Group (DRWG), which has never been formalized, and its FITS Special Expert Group (FITS SEG), which has never been convened in its post-IAU FITS Working Group composition, before the next IAU General Assembly in 2021. There could be two additional SEGs: one for Structured Data Formats such as ASDF and HDF, and another for Virtual Observatory Data Formats in conjunction with the IVOA. Jessica Mink is chairing the DRWG, and Lucio is chairing the FITS SEG for now.

4. Discussion

We are missing models and vocabulary that can be serialized. Machines can open FITS files, but can't understand them except for the standardized world coordinate system (WCS). We only have a basic format definition. A file cannot tell you in a standard way what convention it is using. There should be keywords that tell you the conventions used and give you the parameter definitions or point to a source for them. We need to keep track of all the keyword definitions in existing data. Who is going to maintain them? The IAU? The central FITS site? Should each instrument team publish their definitions and formats in refereed data papers to circulate them widely? It was noted that data models are all about that. You have a complete mapping of the data into the model.

Should the IAU Data Representation Working Group propose the standards? Why involve the IAU when we already have standards? We want to maintain longevity of the exchanging format as we have with FITS through the IAU FITS Working Group and publication of FITS standards in refereed astronomical literature. It was suggested that the IVOA does not have enough processes to do the second, though it might also be said that the standards are only just now reaching a stability which allows for publication in a refereed journal. The IAU and IVOA are discussing ways to work together to maintain the standards into the future.

It was suggested that we should talk to oceanographic data specialists about their standards as they seem to do it very well.

The problem we are seeing over the years is that we know we need to change the standard but the community as a whole does not want to change them.

Anne Raugh noted that while the PDS works with metadata, old data often doesn't have metadata recorded. We should also worry about endangered datasets. The PDS management wants to know of any in planetary science.

There was a general consensus of the meeting that we should move to data models, a more transparent markup language such as YAML or JSON, ask IAU to work on metadata standards, and figure out how to maintain keyword definitions.

5. Assignment for Next Year

Over the next year, there will be advocacy for the adoption of long keywords in FITS, including the HIERARCH convention as a subset. Developers of data formats for new instruments are invited to continue to talk among themselves concerning shared conventions.

References

- Berry, D. S., Warren-Smith, R. F., & Jenness, T. 2016, *Astronomy and Computing*, 15, 33. 1602.06681
- Dencheva, N. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 535
- Greenfield, P., Droettboom, M., & Bray, E. 2015, *Astronomy and Computing*, 12, 240
- Jenness, T., et al. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 653
- Marmo, C., Hare, T. M., Erard, S., Cecconi, B., Minin, M., Rossi, A. P., Costard, F., & Schmidt, F. 2018, in *Planetary Science Informatics and Data Analytics Conference*, vol. 2082 of LPI Contributions, 6024
- Mink, J. 2019a, in *ADASS XXVII*, edited by J. Ibsen, M. Solar, & P. Ballester (San Francisco: ASP), vol. 522 of ASP Conf. Ser., 689
- 2019b, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 201
- Morales, F. 2012, in *Astronomical Data Analysis Software and Systems XXI*, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461 of Astronomical Society of the Pacific Conference Series, 685
- Pence, W. 2017, HIERARCH Convention for Extended Keyword Names. Part of CFITSIO documentation on NASA Goddard web site, URL https://heasarc.gsfc.nasa.gov/fitsio/c/f_user/node28.html
- Wicenec, A., Grosbol, P., & Pence, W. 2009a, Registered FITS Convention: The HIERARCH Keyword Convention. Registered convention on NASA FITS web site, URL https://fits.gsfc.nasa.gov/registry/hierarch_keyword.html
- 2009b, The HIERARCH Keyword Convention. Unpublished document available through NASA FITS web site, URL <http://fits.gsfc.nasa.gov/registry/hierarch/hierarch.pdf>



Peter Shawhan leading the “Multi-Messenger” BoF (Photo: Peter Teuben)

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

Data Analysis Challenges for Multi-Messenger Astrophysics

Peter S. Shawhan,^{1,2} Patrick R. Brady,³ Adam Brazier,⁴ S. Bradley Cenko,^{5,2} Mario Jurić,⁶ and Erik Katsavounidis⁷

¹*Department of Physics, University of Maryland, College Park, MD, USA;*
pshawhan@umd.edu

²*Joint Space-Science Institute, Maryland, USA*

³*Center for Gravitation, Cosmology and Astrophysics, University of Wisconsin–Milwaukee, WI, USA*

⁴*Department of Astronomy, Cornell University, Ithaca, NY, USA*

⁵*NASA Goddard Space Flight Center, Greenbelt, MD, USA*

⁶*Department of Astronomy, University of Washington, Seattle, WA, USA*

⁷*LIGO Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA*

Abstract. Recent multi-messenger observations of gravitational-wave and high-energy neutrino sources together with electromagnetic signatures have opened new ways of observing the Universe. These promise a future in which physics and astronomy will be advanced by combining observations and data from across the electromagnetic spectrum with gravitational waves and neutrinos. We consider the challenges the field is facing in fully utilizing data for multi-messenger astrophysics. Such data come from heterogeneous detector networks and standards, and their analysis is often time-critical to guide further observations. In this area, science capabilities depend on the interplay among observation, theory and computational/modeling work. Advances in data science and computing present additional opportunities and considerations in analyzing such data. We invited ADASS participants to a Birds of a Feather session to engage in discussion on the challenges and opportunities in data analysis for multi-messenger astrophysics.

1. Introduction

Until recently, mankind relied almost exclusively on electromagnetic (EM) waves (or photons) spanning the spectrum to reveal the content and workings of the universe. Large, highly sensitive neutrino and gravitational-wave observatories have now made additional astrophysical “messengers” available for study, with initial discoveries (Aartsen et al. 2013; Abbott et al. 2016) progressing to understanding source populations (Aartsen et al. 2018; LIGO Scientific Collaboration & Virgo Collaboration 2018b,a).

Furthermore, highly energetic sources such as neutron star binary mergers, stellar core collapse, and relativistic jets can produce multiple types of emissions. The relatively new field of *multi-messenger astrophysics* involves combining observations of the same events with different messengers, taking advantage of their different characteristics to collect and relate information about the core “engine”, outflows and environment of individual events or of populations. Modern facilities and instruments have strengths

to reveal different properties. For instance, gamma-ray and neutrino fluxes and spectra can reveal particle acceleration; X-ray and radio afterglows can localize sources and characterize their environments; gravitational waves can measure binary mass and orientation parameters and distance; and visible/infrared can precisely localize events, provide redshifts, and trace out expansion and thermal signatures.

The idea for this Birds of a Feather (BoF) session came out of a workshop held at the University of Maryland in May 2018. That Workshop on Cyberinfrastructure for Multi-Messenger Astrophysics (CiMMA 2018) brought together about 20 experts in astrophysics, computational science, and everywhere in between to discuss how cyberinfrastructure—broadly interpreted—can and should be used more effectively to enable multi-messenger science. The opportunities and challenges considered during the workshop are discussed in the workshop report (Allen et al. 2018).

At around the same time, the U.S. National Science Foundation's Office of Advanced Cyberinfrastructure encouraged proposals to its Cyberinfrastructure for Emerging Science and Engineering Research (CESER) program for "scalable data-driven cyberinfrastructure exemplars that will accelerate discovery for one or more science and engineering research communities". We and our colleagues recognized the ideal match to the goals and needs of multi-messenger astrophysics—bringing together two of NSF's Big Ideas—and launched the SCiMMA Project: Community planning for Scalable Cyberinfrastructure to support Multi-Messenger Astrophysics.¹ The ADASS BoF session provided an ideal venue to share this vision and gain valuable input from experts in attendance.

2. Science Drivers

The 2017 gravitational-wave detection of the binary neutron star merger GW170817 (Abbott et al. 2017a), accompanied by a gamma-ray burst and followed up thoroughly using EM facilities around the world (Abbott et al. 2017b), was an outstanding inauguration of multi-messenger astrophysics which yielded many important insights. However, it is just one example. The following scenarios, which could plausibly arise over the next several years, give a sense of other science opportunities:

- Early warning of a nearby compact binary merger, via gravitational waves, allows the earliest phase of the EM counterpart to be identified and measured.
- High-energy neutrinos detected and localized to a galaxy cluster trigger EM follow-up and the observation of a tidal disruption event.
- Gravitational waves from supermassive black-hole binaries detected by pulsar timing arrays allow studies of galaxy properties and accretion disk physics.
- A Galactic or Local Group supernova is observed in all the messengers!

Initial detection of these events will likely be done by "anchor" facilities such as LIGO, IceCube, wide-field gamma-ray survey missions, LSST, Super/Hyper-Kamiokande, and pulsar timing arrays, but follow-up observations by many other facilities will be crucial to fully understand these systems. Efficient, robust searches for signals must be supplemented with rapid characterization, photometric analysis, signal modeling and multi-dimensional parameter estimation to extract astrophysics from the combined observational data.

¹<https://scimma.org/>

3. Cyberinfrastructure for Multi-Messenger Astrophysics

In this context we consider *cyberinfrastructure* to include distributed data-handling, computing, analysis, and collaboration services/systems to enable discovery, education, and innovation. Of course, many advanced computing systems and techniques are in place to serve individual projects, and various communication and information-sharing tools already exist, such as SNEWS, AMON, GCN, TNS, SNEx, ATel, ANTARES, and VOEvent. Survey data and catalogs of known objects are also important for identifying and classifying transients.

Multi-messenger astronomy, by definition, involves diverse facilities. Doing it well presents distinct challenges due to having highly heterogeneous facilities, data, and people; high-volume, high-velocity transient streams; rapidly developing, dynamic collaborations; heterogeneous data sharing policies; competition for follow-up resources; the need for rapid modeling and intelligent scheduling; and tension between human versus machine communications. To realize the science potential of multi-messenger astrophysics, we expect the community to need capabilities such as:

- a scalable framework to facilitate joint analysis, enabling teams to work together, while respecting different scientific cultures
- real-time decision making in event observation and follow-up
- coordination of observing resources, with capability-based access controls
- sustainable, long-term archival storage with standardization of data sets
- data escrow, pre-registration of analyses
- machine-readable communication standards and protocols

Pursuing this also offers many interesting avenues for research and development of computer/data science, including machine learning (deep learning), purpose-built hardware for real-time inference, data access/analysis with differential privacy, uncertainty quantification and predictive modeling, handling of missing or imbalanced data, and resource optimization.

The general goal of the SCiMMA Project is to identify the key questions and cyberinfrastructure projects required by the community to take full advantage of current facilities and imminent next-generation projects for multi-messenger astrophysics. Using input from the general community and specific stakeholder groups, the project aims to produce a community white paper documenting needs and opportunities, and a strategic plan for an institute to address these needs, by the middle of 2019.

Join the conversation! Visit the scimma.org web site, join the “scimma” Google Groups forum, attend upcoming workshops, and contribute to activity areas of interest to you (data management, application integration, machine learning, planning observations, astrophysical inference, modeling and theory, education and workforce development, etc.) We are endeavoring to understand how this kind of distributed work can be coordinated effectively through an institute to support excellent science in the future.

4. Discussion

Audience questions and discussion in the BoF session provided useful input. Notable lines of discussion included:

- The role of inter-agency task forces or other coordination, within the U.S. and internationally, for setting consistent priorities and complementary funding.

- We should absorb and build on, as much as possible, the activities and lessons learned from the ASTERICS project,² which has been underway in Europe since 2015 and is completing in 2019.
- Consider what systems already exist for communication and coordination of robotic telescopes, worldwide. How can we build on it?
- Some multi-messenger observations are open while others will involve proprietary data. Part of this effort is to understand the policy issues and to enable collaborations to form quickly, avoiding a slow ad-hoc memorandum-of-understanding process while still respecting data access restrictions.
- Besides detecting potentially interesting events, it is important to quickly evaluate them and make judgments about using additional observing resources.
- How do we get relevant people involved, and how can we ensure that useful tools are actually used? Need to engage people and existing systems, and consider the sociological and educational aspects as much as the technical ones.
- How can we work effectively with aggregate data sets which come from different instruments? Do we attempt to host data in one place, or do distributed processing? Can we find and refer to all the relevant data with one handle?
- The ability to filter information streams is important, using “brokers” or other mechanisms.

We look forward to future input and discussion on these and other topics.

Acknowledgments. We acknowledge the participants at the CiMMA 2018 workshop and all contributors to the ongoing SCiMMA Project for useful discussions, and we thank everyone who attended the ADASS Birds of a Feather session for their input. This planning and community engagement effort is supported by the U.S. National Science Foundation through grants PHY-1838082 and OAC-1841625.

References

- Aartsen, M., et al. 2018, *Advances in Space Research*, 62, 2902 . Origins of Cosmic Rays, URL <http://www.sciencedirect.com/science/article/pii/S0273117717303757>
- Aartsen, M. G., et al. 2013, *Phys.Rev.Lett*, 111, 021103. 1304.5356
- Abbott, B. P., et al. 2016, *Phys.Rev.Lett*, 116, 061102. 1602.03837
- 2017a, *Phys.Rev.Lett*, 119, 161101. 1710.05832
- 2017b, *ApJ*, 848, L12. 1710.05833
- Allen, G., Anderson, W., Blaufuss, E., Bloom, J. S., Brady, P., Burke-Spolaor, S., Cenko, S. B., Connolly, A., Couvares, P., Fox, D., Gal-Yam, A., Gezari, S., Goodman, A., Grant, D., Groot, P., Guillochon, J., Hanna, C., Hogg, D. W., Holley-Bockelmann, K., Howell, D. A., Kaplan, D., Katsavounidis, E., Kowalski, M., Lehner, L., Muthukrishna, D., Narayan, G., Peek, J. E. G., Saha, A., Shawhan, P., & Taboada, I. 2018, *ArXiv e-prints*, arXiv:1807.04780. 1807.04780
- LIGO Scientific Collaboration, & Virgo Collaboration 2018a, *arXiv e-prints*, arXiv:1811.12940. 1811.12940
- 2018b, *arXiv e-prints*, arXiv:1811.12907. 1811.12907

²<https://www.asterics2020.eu/>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Beginners Guide to Machine Learning in Astronomy

Kai Polsterer and Nikos Gianniotis

*Heidelberg Institute for Theoretical Studies gGmbH, Astroinformatics,
Heidelberg, Germany; Kai.Polsterer@h-its.org, Nikos.Gianniotis@h-its.org*

Abstract. Machine learning has become a key tool to analyze and process complexly structured large datasets. This BoF will be different than the usual BoFs, with the aim of discussing a specific topic. Due to the large request in understanding and learning machine learning techniques at the previously organized ADASS, we will take the opportunity to introduce basic concepts of machine learning. Based on a few examples, different machine learning models will be introduced and their application will be shown. At the end of the BoF, participants will have a basic understanding of what machine learning is about. To enable the participants to further learn about machine learning and to allow for a broader overview, a list of good online-sources will be provided.

1. A Guide to Estimating Redshift and Beyond

(1) Purpose. The present document should be read in conjunction with the online available notebook that was presented during ADASS 2018. Sections are numbered according to the sections in the notebook. The purpose of this document is to supplement the online material with brief explanations and additional bibliographical references.

(2) Running examples: star-galaxy classification and redshift estimation. Photometric redshift estimation is a particularly important task in astronomy as this allows astronomers to infer cosmological properties of the universe. For instance, estimating the redshift of quasars imparts us with information on the evolution of the early universe. To that end, spectroscopic surveys have been employed, which though extremely precise in determining redshift, they are also extremely time-intensive and cannot be used to study large fractions of objects. Nowadays, however, it is the data-driven approaches that are constantly gaining ground. Roughly speaking, such methods approach the task as a regression problem where the inputs (i.e. independent, features, or predictor variables) are physical quantities derived from observed photometry and the output (i.e. dependent, response or predicted variable) is the previously estimated redshift. Hence, photometric redshift estimation constitutes a realistic example for introducing the elemental concepts of machine learning.

(3) Predictive models. One of the quests of machine learning is delivering good predictive models: given previously gathered observed data, for which ground truth is available in the form of a class label (classification) or a numerical target (regression), we would like to build a model that delivers good predictions for new incoming data

(i.e. previously unseen data). For instance, given observations from a new celestial object, predict whether the class label is a star or galaxy; given new photometric features predict the numerical value of the redshift. A good predictive model is one that generalizes well. This means, based on a dataset (the training set), we would like the predictive model to identify the underlying relation (e.g. physical law) which connects the inputs to the targets. This stands in contrast to the predictive model identifying spurious relations, or relations that hold only on the training set, but do not apply in general on yet unseen data. Of course, new unseen incoming data are not available when building the predictive model. We will see later, however, that the situation of evaluating on unseen data, can be “simulated” via the use of testing datasets. When a predictive model performs well on the training set but does not generalize well, one says that the model overfits the training data (Bishop 2006, Chapter 1).

(4) Classification. We briefly dwell on an example of a classification rule that decides whether an observed object is a star or a galaxy. Early approaches, relied on simple criteria such as the difference between magnitudes. While such criteria are easy to understand, they do not make adequate use of the available information present in the data.

(5) Classification with reference data. Data driven methods build a model by making use of a dataset. A powerful algorithm is the K-nearest neighbor algorithm, a classification algorithm (Mitchell 1997, Chapter 8). The algorithm operates as follows: in order to determine the class of a given data item, it looks at the labels of its K closest neighbors, i.e. the K closest data items in Euclidean distance. The most frequent class label of the K neighbors is assigned to the data item in question. Using two features (like in the examples) allows us to visualize the classification boundaries that separate the classes. However, nothing prevents us from using more features or combinations thereof.

(6) Evaluation. We need criteria to quantify the performance of the classifier. Such criteria are the true/false positive/negative rates which inform us of the type of mistakes the classifier commits (Fawcett 2006). Such a criterion becomes especially important in a classification setting of multiple classes as it may happen that the classifier performs well on a number of classes but poorly on a particular class. A tool for diagnosing the behavior of the classifier is the confusion matrix.

(7) Validation. As aforementioned, we wish to build a predictive model that generalizes well and does not overfit (Bishop 2006, Chapter 1). In other words, the model should perform well in general on data we may encounter in the future, and not just on the dataset that we happen to possess. Though we only possess a single dataset, we can simulate the situation of receiving yet unseen data on which we can test how well the classifier generalizes. One way of doing this is by randomly splitting the dataset into two distinct, non-overlapping partitions, a training set and a testing set. One then uses the training set to build the predictive model (i.e. train it) and the testing set to evaluate it using the aforementioned predictive criteria. The testing set simulates the new, yet unseen data that the predictive model may encounter during its deployment. One can improve upon this method, by repeatedly splitting the dataset. This allows us to obtain a more robust evaluation that is not dependent on a single random partition of the

data. Instead, cross-validation partitions the dataset into K non-overlapping partitions. Subsequently, each of the K partitions is designated as the testing set, while the other $K - 1$ become the training set that is used to build the predictive model. The average of the K performances on the testing set quantifies the generalization performance of the predictive model (Hastie et al. 2009, Chapter 7).

(8) Unbalanced training data. An aspect that one has to pay attention to is class unbalance. This means, that a particular class may be severely outnumbered by other classes. Such an unbalance may make the training of a classifier difficult. An extreme case helps understand the issue: the classification task consists of a minority class with very few members and a majority class that makes up almost all the data. A classifier that would constantly output the label of the majority class would seem to perform well (according the aforementioned criteria). Common ways of circumventing this problem is by either populating the minority class with additional (synthetic) data items or by removing data items from the majority class (Chawla et al. 2002).

(9) Regression with reference data. This section shows how the K -nearest neighbor algorithm can be employed in a regression setting. Similar to the classification setting, the target (numerical label) of a data item in question is estimated by looking at the targets of its K closest neighbors. The target is estimated as the average of these K values, though one can think of other ways of weighing the K values (Kuncheva 2004) and obtaining a prediction.

(10) How to further optimize - other algorithms / decision trees. We always wish to make use of all the available information present in a dataset. To do so, we need to adapt our design choices to the dataset at hand. This may involve, for instance, choosing features, experimenting with alternative distance functions, preprocessing the data in different ways and tuning hyperparameters. However, one should be aware that every time a choice is made that adapts to the data one runs the risk of overfitting, that is, overly adapting to the data so that generalization performance is compromised. This makes the use of cross-validation imperative as it helps to prevent harmful design choices that do not uncover the true, underlying relations of the data, but instead lead to overfitting.

(11) Ensemble learning - random forest. A useful method for producing predictive models that are robust to overfitting is ensemble learning (Breiman 1996). The main idea is to use a set of models for making predictions as opposed to a single one. In a classification setting, one can train multiple classifiers and then output a single prediction by a majority vote; in regression, one can train multiple regressors and then output a single prediction by averaging the individual predictions. Roughly speaking, the intuition is that, if the predictive models are constructed independently of each other, then they will commit independent errors (Sammut & Webb 2010). Hence, when combining the predictions, the independent errors will cancel each other out. The random forest is a prominent example of ensemble learning (Breiman 2001). The advantages of ensembles extend beyond warding off overfitting as they can help expand the hypothesis space dictated by the predictive model (Dietterich 2000) or provide mechanisms for handling large datasets (Chawla et al. 2004)

(12) Doing science with all the things presented above. Most of the discussed concepts are exemplified in the astronomical task of predicting photometric redshifts in (D’Isanto et al. 2018). Therein, a method for massive feature selection is discussed that discovers a set of powerful features capable of boosting predictive performance.

(13) Artificial Neural Networks. Like K-nearest neighbor, artificial neural networks (ANN) (Bishop 2006) can serve in both regression and classification settings. ANNs are realized as parameterized functions that learn an explicit mapping between the inputs and targets, as opposed to the implicit mapping determined by K-nearest neighbor. A large number of techniques falls under the category of ANNs including autoencoders, RBF networks, recurrent neural networks, convolutional networks and deep networks to name just a few (Goodfellow et al. 2016).

(14) What’s next. The presented BoF has only touched upon the most elemental concepts of machine learning. The literature is vast and specialized as to appear bewildering to the uninitiated. Fortunately, there are number of useful books dedicated to the subject that address newcomers but also more advanced learners that wish to revisit certain topics. Among them, we only mention but a few Mitchell (1997), Bishop (2006), Hastie et al. (2009), Murphy (2012), Barber (2012), Goodfellow et al. (2016), though obviously alternatives may be better suited depending on the level and personal preferences of the reader.

Acknowledgments. The authors acknowledge the generous and invaluable support of the Klaus-Tschira Stiftung.

References

- Barber, D. 2012, Bayesian reasoning and machine learning (Cambridge University Press)
- Bishop, C. M. 2006, Pattern Recognition and Machine Learning (Springer)
- Breiman, L. 1996, Machine Learning, 24, 123
- 2001, Machine learning, 45, 5
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002, Journal of artificial intelligence research, 16, 321
- Chawla, N. V., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. 2004, Journal of Machine Learning Research, 5, 421
- Dietterich, T. G. 2000, in International workshop on multiple classifier systems (Springer), 1
- D’Isanto, A., Cavuoti, S., Gieseke, F., & Polsterer, K. L. 2018, A&A, 616, A97
- Fawcett, T. 2006, Pattern Recognition Letters, 27
- Goodfellow, I. J., Bengio, Y., & Courville, A. C. 2016, Deep Learning (MIT Press)
- Hastie, T., Tibshirani, R., & Friedman, J. H. 2009, The elements of statistical learning: data mining, inference, and prediction, 2nd Edition (Springer)
- Kuncheva, L. 2004, Combining pattern classifiers: methods and algorithms (John Wiley & Sons)
- Mitchell, T. 1997, Machine Learning (McGraw-Hill)
- Murphy, K. P. 2012, Machine Learning: A Probabilistic Perspective (MIT Press)
- Sammut, C., & Webb, G. I. (eds.) 2010, Negative Correlation Learning (Springer US), 715

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Data Citation: from Archives to Science Platforms

August Muench¹ and Raffaele D'Abrusco²

¹AAS, Washington, D.C., USA; august.muench@aas.org

²Center for Astrophysics | Harvard & Smithsonian, Cambridge, MA, USA

Abstract. As data citation and data preservation have gained increased prominence in recent years, we feel that it is important to continue the conversation that was started during the DOI BoF at ADASS XXVII and expand its scope. To be more specific, some of us, as data editors of the AAS Journals, see steady growth in authors using both high-level science and/or multi-purpose generic repositories for archiving publication related data & related materials. Data providers as well have acknowledged the need to adopt global, metadata-rich, persistent identifiers for data products they collect, curate, and share. We also anticipate an increase in funder or journal policies that encourage/require authors to make such data persistently available at publication. The BoF discussion included presentations and summaries of how astronomy and planetary archives are supporting data citation today.

1. Introduction

This Data Citation Birds of a Feather (BoF) session built upon a BoF organized by A. Rots for ADASS XXVII (Santiago, Chile) and a followup splinter meeting during the January 2018 meeting of the American Astronomical Society (#231). Rots et al. (2018) provides a introduction to some of the issues discussed in these prior sessions, which focused upon ways to map *Chandra* or other observatory and Virtual Observatory metadata into the DataCite Digital Object Identifier (DOI) metadata schema (V4.1+).

We expanded the scope of 2018 ADASS Data citation BoF in two ways. First, we set a goal to document the growing or planned usage for data-related DOIs by astronomy data centers. Presentations were solicited from a number of organizations to explain their perspectives on minting DOIs to data. Second, we included the future growth of “Science Platforms” as compute centers for astronomical data. Researchers using these centers will want to persistently link to the data they produce and should be able to transmit credit for the software and catalogs that underlie these platforms.

1.1. Publisher Perspective

The most common reason publishers want data DOIs is to establish a persistent link between articles and the data used in or created for that article. The AAS Journals and NASA ADECs have undertaken work on linking data to articles before (Schwarz 2005). The main change from the original dataset tags, which were not widely used by authors, is employing a broad, community oriented initiative (run by CrossRef and DataCite) that collects and retains metadata about the link to improve persistence. Such a system is easier to use and maintain than archive specific identifiers.

Similar changes are happening in research fields adjacent to astronomy. The Journals of the American Geophysical Union are adopting a FAIR data requirement: ¹ data used in an AGU Journal paper must be available at publication, and cannot be stored with the Journal. While websites are being developed to direct authors to partner repositories, it is hard to find pre-publication astronomy-oriented data centers, and the AAS Journals are concerned with directing authors primarily to “generic” repositories, e.g., Zenodo, which lack domain-specific interoperability.

2. Presentations

2.1. The *Chandra* Data Archive and PIDs

The *Chandra* Data Archive (CDA) has long been interested in using permanent identifiers (PIDs) to link between Journals and data, participating in the creation of the Journals/NASA ADEC PID scheme 16 years ago. The CDA is thus migrating to utilize DataCite DOIs. There is a strong self-interest to show that *Chandra* maintains value as it ages, but for a migration to DataCite, maintenance of DOI metadata and the DOI landing pages need to be carefully considered given the transition to the permanent HEASARC repository after the close-out phase of the mission. Maintaining continuity across that transition is a big commitment and challenge.

The CDA plans to mint three types of data DOIs: single-observation; observational aggregates; and user-contributed datasets, which are similar to HST HLSPs but *Chandra*-specific. One of the more challenging pieces is versioning of DOIs as the underlying calibrations are updated and the data reprocessed. The *Chandra* Source Catalog may also mint DOIs too, but there are open questions, regarding the granularity level of the DOI and the infrastructures that would allow user to resolve the basic units used in their research (single sources, heterogeneous groupings of sources, data products, etc). While minting a single DOI for each release of the CSC is reasonable, striking a balance between full reproducibility and technical feasibility is a big undertaking.

2.2. Report from IVOA Data Curation and Preservation (DCP) Interest Group

A. Schiff reported on the IVOA Data Curation and Preservation Interest Group Session that took place on November 10, 2018². A few take aways: granularity (as discussed for *Chandra*) requires careful thought. DOIs are not always necessary but the DataCite DOI system can be used to helpful usage stats where a plain archive PID may not. It is potentially hard to figure out which metadata to require/allow of an author/archive: should an archive issue DOIs for entities that exist or are defined externally (e.g. Gaia catalog mirrored by many archives). If so, should archives strive to make the connections between different DOIs pointing to the same data objects? Luckily, the rich set of attributes available in the DataCite DOI metadata scheme allows such connections. The real question is: do all the entities involved in the game of minting DOIs agree to setting common rules? The IVOA DCP will be drafting an “IVOA Note” on data DOIs.

¹FAIR: <https://www.force11.org/group/fairgroup/fairprinciples>; AGU Enabling FAIR: <http://www.copdess.org/enabling-fair-data-project/>; Repository Finder: <https://repositoryfinder.datacite.org/>.

²See the IVOA DCP session page on Data DOIs for additional materials: <https://wiki.ivoa.net/twiki/bin/view/IVOA/InterOpNov2018DCP>

2.3. CADC/CANFAR

CANFAR³ is a Canadian science platform that has been operating since 2011. Their data DOI archiving workflow derives from existing infrastructure for user storage, which has a VOSpace heritage (Kavelaars et al. 2012). CANFAR is authorized by DataCite Canada to issue DOIs via a manual process of “freezing” the users’ data and submitting the DOI metadata to DataCite. No metadata is collected besides the paper itself. Users can use whatever data (file) structure and documentation they want. Quotas are up to 5 TB, which will be maintained into perpetuity; this includes updating statically generated DOI landing pages with direct links into the underlying data repository.

This simple but manual process is still too heavyweight for the CANFAR team, and they are building a new user-driven workflow. Users will make requests, upload data, edit metadata to mint DOIs themselves. Metadata are still basic: title, author(s), journal ref (text or DOI). The goal is to provide a very generic place for people to upload their data and not a structured telescope archive. CANFAR is thinking about adding support for user-created databases as an increasingly common use case. Such databases could potentially provide TAP access after publication, which would make them more inter-operable than simple data storage.

3. Discussion

More on CANFAR: There was an extended discussion on the CANFAR effort, beginning with how authors move data across the system. S. Gaudet reported that the currently manual process is expected to become automatic in the new user-driven UI. There were questions about how datasets are linked to outside resources: CANFAR DOIs do not appear in the VO registry as the Journal article is the intended discovery vehicle. A follow-up concerned how linked is the paper to the data? The paper-data linkage is about specifying that data is a supplement to the article; a separate paper citing the same data is a different kind of relationship. S. Gaudet also noted that CANFAR can make the data available to reviewers during peer-review.

Planetary Data Service, Small Bodies Node: A. Raugh reported that the Planetary Data System / Small Bodies Node is now a DOI-issuing member of DataCite. PDS’s goals in minting DOIs include providing unique forms of credit to the full list of contributors by role (programmers, archivists) for the data. PDF datasets are Federal records, mandated to be permanent and immutable, which is also a great match to DOI. PDS has a strong technical infrastructure to preserve and manage updates to databases, and is looking to the MAST model for data “slicing” and aggregation. Lastly all PDS datasets are refereed, which may mean that PDF DOIs should be treated differently by authors and publishers than, for example, MAST DOIs.

MAST: T. Donaldson discussed MAST’s data DOI effort (Novacescu et al. 2018), and the social aspect of when to require the creation of data DOIs and when to ask nicely. MAST and the AAS Journals have run a prototype project for over a year, asking STScl scientists to report on and mint DOIs to MAST data if used. Anecdotally, good adoption rates are observed when authors are asked, and the Journals are interested in expanding the number of authors and archives involved.

³http://www.canfar.net/en/docs/digital_object_identifiers/

Others: G. Landis reported that Vizier is planning to mint DOIs for their catalogs (1 DOI per catalog; not per table). They see author needs for pre-publication access to data archiving services but worry about temporary DOIs in the literature. Other DOI services mentioned include Leibniz Institut AIP, NIST, and multiple ANU/CSIRO efforts.

3.1. Science Platforms

At least three “Science Platforms” were present at ADASS: the NOAO Data Lab, LSST, SciServer⁴; members of the LSST team were able to attend the BoF. DOIs for database queries is a topic being discussed at LSST, but data rights are also a factor when query DOIs could be used by non-LSST scientists. T. Jenness inquired about how far down the derived data products rabbit hole would we want to go beyond SQL queries. Should we record what users do when they filter down a billion-row query? It could be argued that the most valuable data are those authors create new from such queries and constitute value-added joins into the LSST data store. The AAS Journals are interested in more than seeing data used or created on science platforms captured and cited in refereed articles. Beyond reproducibility, Journals desire that platform data citation is “transitive” (Katz 2014), acknowledging the software, services and data used on them. G. Dubois-Feldman pointed out that an important issue for LSST and perhaps for all science platforms is that they do not have a preservation mandate from their funding agency (NSF for LSST). Agencies might need to be told that they need to support preservation in the context of projects like LSST or any of the agency funded archives.

4. Next Steps

The BoF spurred followup conversations at ADASS, including meetups with SciServer and MAST, focusing on models for how their users could wrap up the data queried and Jupyter Notebooks used into a single citable unit. New efforts by IPAC for minting DOIs to high-level data products were detailed. Another data DOI splinter meeting at the January 2019 AAS #233 meeting (Seattle, WA) is planned. Last, an AstroDOI email list is maintained by A. Rots; email arots@cfa.harvard.edu to ask to be added.

Acknowledgments. We would like to thank Arnold Rots for discussions setting the scope of this BoF, and Peter Williams for note-taking during the BoF. Tom Donalson summarized the BoF during the last session of ADASS XXVIII.

References

- Katz, D. S. 2014, *Journal of Open Research Software*, 2, e20. URL doi:10.5334/jors.be
- Kavelaars, J., Dowler, P., Jenkins, D., Hill, N., & Damian, A. 2012, in ADASS XXI, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461, 367
- Novacescu, J., Peek, J. E. G., Weissman, S., Fleming, S. W., Levay, K., & Fraser, E. 2018, *The Astrophysical Journal Supplement Series*, 236, 20
- Rots, A., D'Abrusco, R., & Winkelman, S. 2018, in *European Physical Journal Web of Conferences*, vol. 186, 12011
- Schwarz, G. J. 2005, in ADASS XIV, edited by P. Shopbell, M. Britton, & R. Ebert, vol. 347, 375

⁴<https://datalab.noao.edu/>; <https://ldm-542.lsst.io/>; <http://www.sciserver.org/>

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
©2019 Astronomical Society of the Pacific

Unconference BoF Session: I Want to Talk About...

Alice Allen^{1,2}

¹*Astrophysics Source Code Library; aallen@ascl.net*

²*University of Maryland, College Park, MD, USA*

Abstract. ADASS Birds of a Feather (BoF) sessions usually focus on one topic that is proposed before it is accepted. This BoF used an “unconference” format, in which topics for discussion are proposed by attendees before and during the conference, and then are chosen by participants at the beginning of the session. Small-group conversation at round tables allowed discussion of numerous topics, with participants self-sorting themselves into discussion groups. The session closed with one-minute reports from each table. The session was also quickly summarized, along with the other BoFs, in the final plenary session.

1. Introduction

The intent of this Birds of a Feather (BoF) session was to allow time for small groups of people to discuss issues of interest to them that either had not come up previously in the conference or not been covered in the detail they desired, perhaps due to time constraints. Ideas to discuss at this BoF were gathered both online in a Google doc¹ and in person (on a sign-up sheet near the registration desk) before and during the meeting, up until the time of the BoF session. The room for this BoF was set with eight round tables that each seated 10 people, each suggested topic was assigned to a table, and participants chose which discussion they wanted to join.

2. Proposed and Selected Topics

The topics proposed for discussion are listed below, with those in *italics* being the topics that people chose to discuss:

- *Improving ADASS*
- *Spectroscopic visualization tools in browsers*
- *Code of conduct and ethics*
- *ADASS prize for software contributions to astronomy*
- How to get ALL the data for particular sky location?

¹<https://tinyurl.com/adassunconference>

- Improving the ASCL
- Software citation and software services

3. Improving ADASS

The five people who discussed ways to improve ADASS were Marc Pound, Erik Deul, Sebastien Derriere, Christophe Arviset, and Nuria Lorente, who took notes. The first question pondered by the group was “*If you could change one thing about ADASS, what would it be?*” This prompted a long list of possible changes, among them greater diversity in the attendees and greater visibility of ADASS within the astronomy community, more interactivity during the conference rather than passive listening, and links in the program to presentation slides and posters, and subsequent questions, such as “*What steps should ADASS take to improve diversity and inclusion?*” and “*Can we experiment with the flipped classroom model?*”

The question as to how ADASS can improve diversity and inclusion generated useful, actionable suggestions, such as partnering with groups that are ahead of ADASS in diversity and inclusion, such as the University of Maryland Center of Women in Computing², to raise the profile of our field to people of color, women, and minorities. Also discussed was the need for each of us to be aware of the impact we each can make on this by, for example, approaching individuals to encourage them to apply for talks or submit proposals for BoFs for posters. The raised question “*Should we have invited talks by people working to increase involvement of underrepresented minorities?*” was answered by an emphatic “*Yes!*”

There was discussion about how full the conference is; this is seen as a good thing, though there is concern about the lack of “down time” as the number of sessions and activities, such as BoFs, has increased over time. Adding more time to the conference without adding more sessions/activities was proposed, to space events out and provide more room for informal conversations; the suggestion was made to survey ADASS attendees live at the conference about lengthening the meeting, which was done on Thursday with a show-of-hands query.

The group also talked about increasing interactivity and active discussion at ADASS, and raising awareness about ADASS overall. The question “*Can we experiment with the flipped classroom model?*” led to suggestions for how this might be done, perhaps with the POC choosing an invited talk to be recorded and available via YouTube for people to watch before ADASS, thus allowing discussion during the allotted conference time. Panel discussions also came up and were not shown much love, as they limit the voices that speak. BoF-style sessions were preferred, and someone mentioned that a fishbowl panel discussion format³ could also increase interactivity. Livecasting the talks to increase interest in the parts of the astronomy community that don’t know ADASS yet was suggested, as was pushing the data analysis side of the conference more to broaden participation.

²<https://mcwic.cs.umd.edu/about>

³<https://qz.com/work/1105697/for-better-conference-panels-try-the-fishbowl-format>

4. Spectroscopic Visualization Tools in Browsers

Erik Tollerud, Iva Momcheva, Keith Shortridge, Kyle Kaplan, and Wes Ryan talked about these tools; they did not reach any conclusions except that there are various tools out there.

5. Code of Conduct and Ethics

This roundtable had four participants: Kimberly DuPrie, who took notes; Sankalp Gilda; Ranpal Gill; and Kai Polsterer. Their wide-ranging discussion included not only codes of conduct, but also the need for diversity on planning committees, and in session chairs and presenters, and the importance of being sensitive to those of differing abilities.

They considered the utility of a code of conduct in providing a mechanism for reporting unprofessional behavior, the need to identify specific people to contact if someone behaves inappropriately, and that repercussions should be clearly stated. They also discussed the importance of diversity and inclusion not only in planning committees but also as a help in encouraging diversity in presenters.

This group was evenly divided for what to do if someone treats another disrespectfully in private (such as disparaging the other). The members of the group recognized that harsh words can adversely affect someone even if it is done in private and should not be tolerated, but on the other side, “we shouldn’t control everything that happens at the conference.” This was phrased as “How is it different if someone privately tells me I’m an idiot at the conference versus them telling me the same thing outside of the conference?”

Among the list of ideas and recommendations generated were:

- Look at the LSST code of conduct⁴ as a good example.
- State in code of conduct that invited speakers, etc. is based on merit.
- Third parties should report inappropriate behavior.
- Put information about the code of conduct in the meeting venue and let people know where to find the entire code online.
- To help with speaker diversity, if an invited speaker cannot attend the conference, see if there is something we can help them with, such as dependent care, to make it more feasible for the person to attend.
- Provide guidelines ahead of time to remind presenters to be aware of differing abilities, such as color-blindness and difficulty in reading small print, when designing their posters or slides.

⁴<https://project.lsst.org/meetings/lst2018/meeting-code-conduct>

6. ADASS Prize for Software Contributions to Astronomy

Four people discussed the idea of ADASS offering a prize for software contributions to astronomy: Omar Laurino, Steve Crawford, Mike Fitzgerald, and this author. The prompt for this roundtable was “*There is no prize for software in astronomy; what needs to be done to present one?*” Unsaid at the beginning of the discussion was that this idea has been discussed within the ADASS Program Organizing Committee off and on for about a year, so both Fitzgerald and Allen had a different baseline for the discussion, which should have been shared initially, than Laurino and Crawford did. This author has learned her lesson and will do better in the future.

Various avenues for bestowing such an award were suggested, one of which was having the AAS, which administers many awards but none for software, manage this award; the idea was that because the AAS has a higher profile than ADASS, the award would be considered more prestigious if coming through that organization. AAS also has experience in administering awards, and this would leverage that organization’s experience. This suggestion met with some resistance; since ADASS focuses on data and software, a software award coming from it would truly be coming from one’s peers. Other objections were that AAS is US-centric and ADASS is an international organization, and offering such an award could serve to raise ADASS’s profile within the astronomy community. The group also talked about the possibility of supplementing the funds ADASS would make available for this award with funds from a funding organization.

During the session reports at the end of the BoF, Kai Polsterer corrected this group’s discussion prompt, stating that there is at least one award for software in astronomy, awarded by the German astronomical society (Astronomische Gesellschaft).⁵

7. Conclusion

The roundtable discussions were stopped after about an hour, after which a representative from each group briefly reported the essence of its conversation to all. A few comments were shared as to whether to repeat this or a similar “unconference” session in the future. Though the number of people who participated was small in comparison to total attendance, the general feelings about this session amongst participants were positive.

Acknowledgments. My thanks to the attendees for their comments, ideas, and advocacy, to the ADASS POC for accepting this experimental BoF, and to Nuria Lorente and Kimberly DuPrie for their excellent notes.

⁵http://www.astronomische-gesellschaft.de/en/activities/press-releases/software-development-award-2018?set_language=en

Session XVII

Ancillary Meetings

Astronomical Data Analysis Software and Systems XXVIII
ASP Conference Series, Vol. 523
P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.
 ©2019 Astronomical Society of the Pacific

The ADASS Time Domain Astronomy Hackathon

Brian Thomas,¹ Alice Allen,² Marc W. Pound,² and Peter Teuben²

¹*Office of Chief Information Officer, NASA HQ, Washington DC*
brian.a.thomas@nasa.gov

²*Astronomy Department, University of Maryland, College Park*

Abstract. In this paper we describe the ADASS XXVIII hackathon, the first associated with an ADASS conference. We provide our motivation, details of the event, and discuss lessons learned. We believe a hackathon associated with the annual meeting has strong positive value for ADASS and should be considered for future events.

1. Introduction

A hackathon seeks to draw together a large group of folks for an intense and extended period of creative programming. Hackathons may be held for a variety of purposes including, but not limited to, teaching (Huppenkothen et al. 2018), to draw together a technical community as a social event (Kellogg et al. 2018), and to draw attention to solving particular challenges or themes (as found, for example, on popular sites such as Kaggle¹). Pa Pa Pe Than et al. (2018) provides a broader overview of hackathon applications and uses.

Our motivation for holding a hackathon associated with the ADASS XXVIII meeting was aligned with outreach to interested individuals; we wanted to highlight topical technical problems that the ADASS community might be concerned with and introduce a new generation of rising computer programmers and scientists to the excitement of solving them. We chose the topic area of Time Domain Astronomy (TDA) to focus on for this event as it was also one of the themes for this year's ADASS meeting (Nebot 2019) and aligned well with the interests of the Department hosting the hackathon. We allowed a loose definition of TDA, dealing with any astronomical data where time was a parameter. Thus projects for this hackathon could involve, for example, variable stars, exoplanets, and bodies in the solar system.

2. Event Organization

The ADASS hackathon took place the weekend before the ADASS starting on Saturday morning and ending at noon on Sunday with the total event time being 27 hours. We provided a space in the University of Maryland Physical Sciences Complex (PSC) as well as snacks and coffee. The participants were required to attend the introduction and

¹kaggle.com

be present for final presentations at 11am on Sunday. Otherwise, they could stay in the PSC building or leave as they desired. A cash award (provided by the City of College Park) was available for the top 3 teams with \$500, \$350 and \$150 being awarded to the first, second and third place teams respectively. The winning team was also provided time to present their hack during the ADASS meeting.

We began by having the participants introduce themselves, their backgrounds and interests. We then introduced the participants to the field of TDA, providing some general background and challenges in this area. Presentations were given by Charlotte Ward (UMD graduate student), Gerbs Bauer (UMD Research Professor), and Brian Thomas (NASA). We highlighted some datasets which could be applied to solving aspects of the challenges. This was followed by a freely flowing brainstorming session where people could discuss ideas and questions, and potential hacks could be focused. Ideas were placed on sticky notes on a wall. Participants were then allowed a short period of time to form teams and brainstorm. After another hour or so, each team presented an outline of their hack, potentially allowing members to join another team if skill sets were better suited elsewhere. In our case nobody decided to join another team.

We allowed for a range of project types. Projects could be new analyses or approaches or novel ways of understanding existing solutions or problems. The final product could be a proof-of-concept app, a plugin to existing code, a storyboard design, or really anything that embodies creative hacking around the TDA theme. We did not require that the final project be polished; a good idea that was well fleshed out could also be submitted. A final presentation of a few slides describing the work including the motivation and approach was the only requirement for consideration for a prize.

We used Devpost² to help structure the hackathon. This site served as a centralized location from which information could be disseminated including rules of conduct and a discussion board which we used to distribute ideas and answer participant questions. Hackathon rules can be summarized as follows:

- Each participant belongs to one team and one final submission, but is allowed to switch teams. Team makeup is not final until the presentation. The maximum team size was 5.
- Only 1 submission per team.
- A Code of Conduct. We did not tolerate harassment of hackathon participants in any form, including, but not limited to, harassment based on gender identity and expression, age, sexual orientation, disability, physical appearance, body size, race, ethnicity, nationality, religion, political views, previous hackathon attendance, lack of computing experience, or chosen programming language or tech stack. Sexual language and imagery was not appropriate at any point in the hackathon including in software hacks, social media, talks, presentations, or demos.

Hackathon participants violating any of these rules could be sanctioned or expelled from the hackathon at the discretion of the hackathon organizers.

²<https://time-domain-astronomy-6010.devpost.com/>, shortened at <https://tinyurl.com/TDAhack>

3. Participants

Our event was set up as a community hackathon and attracted students, professional hackathonners, and ADASS participants who formed teams (see below). Members of the hosting department and the ADASS program organizing committee served as judges. Out of the 34 original registrations, 6 were present but not playing (being part of the organization or just cheerleading), and 9 did not show up.

Judges, Organizers, and Teams

The session was organized by Peter Teuben, Brian Thomas, Alice Allen, Marc Pound, and Elizabeth Warner. Our judges were Alice Allen, Gerbs Bauer, Andy Harris, Nuria Lorente, Ada Nebot, and Brian Thomas. The 7 teams that participated are listed in Table 1. We have also noted which teams won which prizes.

Table 1. Hackathon Teams	
Team Members	Project Name
Sarah Frail and Patrick Shan	Morpheus - Near Earth Objects Visualization
Marco Lam	Drag and drop ensemble (2 nd Prize)
Paul Ross McWhirter and Josh Veitch-Michaelis	Auto periodogram selection using MC (3 rd Prize)
Timothy Henderson and Matt Graber	Solar Activity Viewer
Thomas Boch, Matthieu Baumann, and Siddha Mavuram	Music of Light curves (1 st Prize)
Kyle Kaplan, Sankalp Gilda, Hayden Hotham, Steve Gambino, and Abbie Petulante	ML on ZTF pipeline
Kevin Cai, Kael Lenus, James Zhou, and Justin Otor	Fixed and Variable Time Kepler Viewer in WWT

The winning team “The Music of Light Curves” made their hack, the sonification of variable stars from the Gaia catalogue, available on <https://github.com/tboch/lightcurves-music>. Their presentation to the ADASS audience during the TDA session on Wednesday met with resounding applause (and later a mention in the international press³).

4. Lessons Learned

As this was the first such event of this type for ADASS we were unsure of the outcome, as it was somewhat of an experiment. We share some lessons learned for future events.

1. *Provide a list of interesting problems and related clean data.* Doing so helps to bootstrap project ideas, as not all participants will have enough domain background to start quickly. Because the event was so short, it was helpful to provide

³<https://www.tomshw.it/altro/sapevate-che-la-luce-delle-stelle-puo-suonare-una-vera-melodia/>

microservices and point to datasets that were more or less cleaned and ‘ready to go’ for projects directed at these problem areas.

2. *Develop a marketing plan.* We could have done a better job to garner interest in the event. We posted to a community BBS, a UMD subreddit, posted paper flyers in campus science and engineering buildings, and contacted student groups and faculty to help spread the word. However, we did not have a coordinated campaign that included social media and messaging targeted for specific dates and groups (e.g., “Save The Date” emails), nor was the hackathon mentioned in the ADASS registration form. A competing, large, all-women hackathon (<https://gotechnica.org/>) held the same weekend on campus also affected our enrollment.
3. *Venue (location and time) is important.* The university was a good choice because of easy access to rooms, wifi, and food choices. Holding the hackathon at a large academic institution ensured that it would be easy for younger participants (undergrads) to attend, as did holding the event over a weekend to avoid conflicting with classes.
4. *Have an assessment tool/strategy.* An exit survey or ending discussion with participants can help improve subsequent hackathons. We failed to take advantage of the opportunity to engage either the participants or the ADASS audience at the session where winning projects were presented about perceived problems and good aspects of our event.
5. *Narrow the range of participant experience.* Future organizers should consider either limiting participation to non-professionals, or group the participants and awards into professional and non-professionals. It is somewhat unfair to have less experienced coders compete against domain specialists and possibly contrary to the avowed desire to use this event to advertise our field of work to outsiders.
6. *Time management is crucial.* Scheduling a conference event right at the end of the hackathon was problematic, and not tightly managing the final presentation time and similar issues became important and detracted from the event. This will be particularly important in other events that have larger participation.

5. Conclusions

A community lives and dies by how well it nurtures the next generation. Folks enter the ADASS community by a number of means but typically by being either scientists who become attracted to the technical challenges of writing the software or as computer engineers and programmers who find the science use cases particularly interesting. We are not aware of any organized means to train the next generation of ADASS workers; there are no formal degree programs in “Astronomy Software.” As such, our community has taken a somewhat laissez-faire approach to training the next generation and this may lead to a future deficit in skilled professionals willing to work in our field. More and more our community’s skills are being found useful in application elsewhere; for example, many ADASS attendees can easily become highly sought after Data Scientists.

Hackathons are a step towards being more proactive in our outreach and provide an ideal means to encourage and interest a younger group of programmers in the complex

and interesting challenges that our community tackles. We found a number of lessons in hosting this event but no showstoppers, and a good deal of goodwill was generated. Based on our experience, we heartily recommend that future ADASS events include hackathon events.

Acknowledgments. We would like to thank the City of College Park⁴ for providing the prize money, Vigilante Coffee⁵ for supplying much needed coffee, ASCL⁶ for providing snacks and the University of Maryland Astronomy Department⁷ for hosting the hackathon.

References

- Huppenkothen, D., Arendt, A., Hogg, D. W., Ram, K., VanderPlas, J. T., & Rokem, A. 2018, Proceedings of the National Academy of Science, 115, 8872. 1711.00028
- Kellogg, L., Hwang, L., Gassmoller, R., Bangerth, W., & Hester, T. 2018, Computing in Science & Engineering, 21, 25
- Nebot, A. 2019, in ADASS XXVIII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 477
- Pa Pa Pe Than, E., Nolte, A., Filippova, A., Bird, C., Scallen, S., & Herbsleb, J. 2018, IEEE Software, PP, 1



Hackathon participants and jurors just before the final presentations. (Photo: Peter Teuben)

⁴www.collegeparkmd.gov

⁵vigilantecoffee.com

⁶ascl.net

⁷www.astro.umd.edu



Most of the hackathon team members. Some of the winners are holding envelopes. (Photo: Brian Thomas)

Astronomical Data Analysis Software and Systems XXVIII

ASP Conference Series, Vol. 523

P.J. Teuben, M.W. Pound, B.A. Thomas, and E.M. Warner, eds.

©2019 Astronomical Society of the Pacific

The International Virtual Observatory Alliance in 2018

Mark G. Allen,¹ Patrick Dowler,² Janet D. Evans,³ Chenzhou Cui,⁴ and Tim Jenness⁵ for the IVOA Executive Committee and Technical Coordination Group

¹*Observatoire astronomique de Strasbourg, UMR 7550, F-67000 Strasbourg, France; mark.allen@astro.unistra.fr*

²*Canadian Astronomy Data Centre, National Research Council Canada, Victoria, British Columbia, Canada*

³*Center for Astrophysics, Harvard & Smithsonian, Cambridge, MA, USA*

⁴*National Astronomical Observatories, CAS, Chaoyang District, 100101 Beijing, China*

⁵*Large Synoptic Survey Telescope, Tucson, AZ, USA*

Abstract. The International Virtual Observatory Alliance (IVOA) held its bi-annual Interoperability Meeting over two and half days prior to the ADASS 2018 conference. We provide a brief report on the status of the IVOA and the activities of the Interoperability Meeting held in College Park.

1. An Alliance for the Global Vision of the Virtual Observatory

The Virtual Observatory (VO) is a collection of interoperating data archives and software tools that facilitate astronomical research. The overall goal is to support innovative research in astronomy by exploiting the full power of growing and emerging datasets and interoperable services. Many projects and data centers worldwide are working to make data and other resources work as a seamless whole. The International Virtual Observatory Alliance¹ (IVOA) is an organization that debates and agrees on the technical standards that are needed to make the VO possible. Constituted in 2002 (see for example Quinn et al. 2004), the IVOA has now been joined by 21 national and international VO projects that meet bi-annually. Major IVOA accomplishments include standards for data and metadata (Data Models), data exchange methods (Data Access Layer; Query Language), and a registry that lists available services and identifies what can be done with them. Organizations have implemented VO-enabled tools and services that can interface seamlessly with VO-enabled archives worldwide, and some projects have built new data systems with VO services at their core. The IVOA acts as a focus for VO objectives, a framework for discussing and sharing VO ideas and technology, and a body for promoting and publicizing the VO.

¹<http://www.ivoa.net>

Recent IVOA activities have focused on the engagement with future large data producing projects, and priorities have most recently focused on multi-dimensional data, and time domain astronomy. IVOA is also evolving as an organization in a rapidly changing landscape, with the emergence of new large initiatives for research data sharing such as the Research Data Alliance (Treloar 2014) with strong support of the **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable (FAIR; Williamson et al. 2016) principles. Astronomy is also changing as we enter an era of very large data, and multi-wavelength and multi-messenger astrophysics where there is an essential need for high level interoperability of data, simulations, tools, and services. The bi-annual Interoperability Meetings are working meetings for making progress on and facing the challenges (Arviset et al. 2017) of coordinating the global effort for interoperability in Astronomy.

The IVOA work is pursued by Working Groups (WG) and Interest Groups (IG) (see Table 1), coordinated by the Technical Coordination Group (TCG), guided by a scientific priorities committee (CSP), with the overall direction provided by the IVOA Executive Committee. Participation in the IVOA working and interest groups is open, as is the participation in the bi-annual Interoperability meetings.

2. IVOA November 2018

The November 2018 Interoperability meeting² was held in College Park, MD, USA, preceding ADASS XXVIII. One hundred and nine participants gathered for two and a half days of productive discussions. Sessions were held by most of the WGs and IGs, running in two parallel streams, and with a number of joint WG-IG sessions.

2.1. Applications

In the Applications (Apps) sessions the themes were: Python and its increasing importance in astronomy software and services; the VOTable format including its use in mapping of data models; HEALPix (Górski et al. 2005) and its use in coverage maps (MOC) and the Hierarchical Progressive Survey (HiPS; Fernique et al. 2015); also the use of Authentication in applications and services. The relationship between Python and the VO was widely discussed and questions were raised about where best to contribute VO tools, how to raise the visibility of Python VO code, and how to avoid duplication. The discussions recognized a need for reference implementations of IVOA standards in Python, and potential follow-up activities such as IVOA hack-a-thons were identified. This is discussed further in Donaldson et al. (2019).

Data Access Layer. The Data Access Layer (DAL) WG sessions included discussions on the current minor version updates being prepared for the Table Access Protocol (TAP), Astronomical Data Query Language (ADQL), and DataLink standards that allow linking of datasets with various resources such as related datasets, metadata, or other services. New proposals were put forward for standards in support of multi-messenger astrophysics. These standards are intended to describe the sky visibility of observatories and missions as is needed for the follow-up of events such as gravitational

²<https://wiki.ivoa.net/twiki/bin/view/IVOA/InterOpNov2018MeetingPage>

Table 1. IVOA Working Groups and Interest Groups

Working Group	Description
Applications	Tools that Astronomers use to access VO data and services. Standards specific to VO Astronomy-user-Applications.
Data Access Layer	Define and formulate VO standards for remote data access.
Data Modeling	Framework for the description of metadata attached to observed or simulated data, and logical relationships between metadata.
Grid & Web Services	Grid technologies and web services within the VO context.
Semantics	Explore technology in the area of semantics with the aim of producing new standards that aid the interoperability of VO systems. UCDs, Standard Vocabulary, exploration of Ontologies.
Resource Registry	Registry provides the mechanism with which users and applications discover and select resources – typically, data and services.
Interest Group	Description
Time Domain	Representation of the emerging time domain community, specific time domain issues in a VO context.
Solar System Theory	IVOA standards in the scope of Solar System sciences. Ensuring that theoretical data and services are taken into account in the IVOA standards.
Education	VO tools, data, and practices in support of astronomy teaching in schools and universities.
Data Curation & Preservation	Share best practices and engage IVOA member projects in the long-term curation and preservation of astronomical data.
Knowledge Discovery	Knowledge discovery is the task of processing and analyzing data-sets with the aim of extracting new knowledge. This area spans visualization, remote data exploration, machine learning techniques, statistical methods, workflow orchestration, and polymorphic data access in the context of the VO.
Operations	Coordinate and publicize activities of individuals, institutions, and groups interested in facilitating robust operations of distributed astronomy applications, particularly those based upon implementations of IVOA protocols.

wave detections. The DAL sessions also included feedback on implementation of VO standards by NED and All-Sky Virtual Observatory (ASVO; O’Toole 2019) groups.

Data Modeling. The Data Modeling WG sessions reviewed the preparation of the major new version of the Space Time Coordinates (STC) standard. Progress was reported on the STC component models: coordinates, measurements, and transformations. A revision of the Simple Spectral Line standard was discussed. The concepts and purposes for the mapping of data models were reviewed. The Provenance data model standards document was in the “Request For Comments” (RFC) phase during the meeting, and presentations on the topic of provenance highlighted the reference im-

plementations using the new model including one being used in the preparation of the Cherenkov Telescope Array (CTA).

Grid and Web Services. The Grid and Web Services (GWS) sessions covered the topic of Authentication and Authorization, in particular the use of OAuth (Open Authorization), an open standard based on tokens. The application of Big Data technologies was discussed, including an example of Apache Spark for simulated Euclid data. Feedback was provided on the use of the VOSpace 2.1 standard. The use of the VO Service Interface (VOSI) for managing versioning of services was discussed, and a new IVOA note is in preparation to outline best practices for its practical use.

Registry. The Registry WG activities were largely combined into joint sessions with other WGs namely DAL, GWS, and Apps. The role of the Registry in providing Authenticated Endpoints for resources was discussed, and a TAP prototype was shown. In the context of the VODataService standard, the use of spatial coverage maps (MOCs) in the registry was discussed. A plan was made to clean up the Registry-of-Registries (RofR) to improve operations by removing invalid records.

Semantics. The Semantics WG reviewed the organization of the various Vocabularies, and discussed the needs expressed by the Theory IG, Data Model WG (for coordinate frame semantics), and Solar System IG. The nomenclature for describing instruments and facilities was also discussed.

Operations. The Operations IG sessions included presentations on service monitoring and “weather reports” of VO services. VO service up-times are shown to be 98-99% overall. VOParis and the European Space Agency (ESA) monitoring systems show that Cone Search services are now generally fully compliant with VO standards. Simple Image Access (SIA version 1) services have some common minor issues, as do the more complex Table Access Protocol (TAP) and Simple Spectral Access (SSA) services. This level of monitoring is very valuable for the operation of the overall system, and also for highlighting areas of the standards that need improvement, indeed some issues may be most appropriately resolved by updating standards. Another topic was HTTPS, reporting that most VO protocols are compatible with HTTPS but challenges remain for WebSAMP (Web enabled Simple Application Messaging Protocol).

Time Domain Astronomy. Time Domain astronomy is a current priority area for the IVOA. The Time Domain IG session addressed the status of the Time Series Data Model, and various ways of expressing time related data in the VO. Following the publication of an IVOA note, a proposal was presented for a TIMESYS element in VOTable, as a basic component for interoperability of time based data expressed in VOTable documents. The driving use case is the combination of two time-series, requiring two time coordinates to be put into a common frame of time scales with a reference position and a time origin of the time scale. Consistency with future STC2 data model is taken into account.

Data Curation and Preservation. The Data Curation and Preservation (DCP) session focused on the role of Digital Object Identifiers (DOIs), including examples of

current practice and different approaches being taken throughout the astronomy and data sharing communities. Vizier presented a proposal for how they were considering using DOIs for their holdings and how this might work for datasets where they were not the primary data source. LSST discussed different ways in which DOIs for queries of large datasets could be issued and how to distinguish between the query itself, for a specific data release, and the results of the query, along with the possibility of allowing curated subsets of query results. Finally, it was agreed to look into issuing DOIs for VO standards. A summary of the discussion was presented at ADASS (Muench 2019).

Solar System. The Solar System IG sessions included reports from projects such as EuroPlanet VESPA, and there was discussion on bridging VESPA and PDS4. The use of the EPN-TAP protocol, a specialized version of TAP used in the planetary science context in the VESPA project, was discussed. A preliminary discussion was held about Space Reference Frames.

Theory. The discussion in the Theory IG session highlighted the need to raise the visibility of Theory IVOA standards in upcoming projects that will make heavy use of simulations (EUCLID, LSST, SKA, etc.), and to make stronger links between observations and theory.

Knowledge Discovery and Education. While the Knowledge Discovery IG and Education IGs did not have sessions at this meeting, the members highlighted the various relevant current topics from across the various WGs and IGs. KDIG emphasized the need for Science Platforms for using and sharing scientific workflows and batch processing using the concept of “code to the data,” and the use of DOIs. Education IG made plans for exploring connections to Education and Public Outreach activities of the IAU for VO-enabled data driven astronomical education.

3. IVOA Next Steps

The work of the IVOA is all aimed at enabling new and innovative science. The IVOA standards, defined *by the community, for the community*, form an important part of the astronomy data infrastructure. Realizing the vision of the VO of course relies on widespread implementation of the infrastructure via the adoption of the standards in data archives, services, and tools. A recent visible success is the publication of the ESA Gaia mission DR2 via VO technologies, proving VO capabilities for handling the peak load challenges of a very large data release (see e.g., Baines et al. 2018; Salgado 2019). The continued growth of the VO system requires that future and current data producing projects and missions are fully engaged with IVOA to ensure that the standards address the changing scientific needs of the community. The IVOA CSP, which has recently expanded with new members has the role of fostering these engagements and identifying the most important scientific priorities to guide IVOA developments.

Astronomy is rapidly entering into a new era of Big Data, and interoperability is increasingly important for multi-wavelength, multi-messenger, and time domain astronomy. The participation of projects such as SKA, CTA, LSST, EUCLID, etc., and the astronomy community in general, is essential to help define the scientific priorities that guide the IVOA activities in these areas. One of the common themes that is being

driven by the needs of these big data projects is the concept of *science analysis platforms* that will enable analysis of the data with capabilities for providing computational resources close to the data. Access to VO resources via these platforms will be essential, and the role of standardization for interoperability of these platforms is a strongly emerging topic being discussed within IVOA. The user-centric focus of the science platforms presents many opportunities to enable a greater level of programmatic access to VO resources, with popular languages and systems used by scientists (e.g., Python notebooks). Other strong themes for future development include the use of machine learning, new visualization capabilities, and use of VO data in education and outreach.

IVOA seeks to welcome new projects and their scientific and technical input. The IVOA web pages and wiki are being updated with information to facilitate the use of VO systems for data providers and astronomers, highlighting the many scientific and practical benefits to publishing data in the VO – that the data become interoperable with other worldwide resources and that the data become visible in many widely used tools. For developers there is information on libraries, clients and tools that can be used to interact with the data when they are available through VO protocols. The revised web pages are also being updated with information on a number of well-tested frameworks available for publishing data.

Following this short-format interoperability meeting in College Park, the IVOA work continues via various email and slack communication channels, using the IVOA wiki for organization of all the activities. The IVOA Newsletters (Baines et al. 2018) provide latest news and results, and a calendar of events. The recently appointed IVOA Media Group continues to enhance the IVOA social media presence, and is making plans for a revised IVOA information web portal. A major 5-day Interoperability Meeting is planned for May 2019 in Paris.

Acknowledgments. The IVOA would like to thank the local organizers of both the IVOA Interoperability Meeting, and the ADASS XXVIII conference.

References

- Arviset, C., Allen, M., Aloisi, A., et al. 2017, in *Astronomical Data Analysis Software and Systems XXV*, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of ASP Conf. Ser., 65. arXiv:1803.07490
- Baines, D., et al. 2018, IVOA Newsletter — August 2018. URL <http://www.ivoa.net/newsletter/019/>
- Donaldson, T., et al. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 237
- Fernique, P., Allen, M. G., Boch, T., et al. 2015, *A&A*, 578, A114. arXiv:1505.02291
- Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. 2005, *ApJ*, 622, 759. arXiv:astro-ph/0409513
- Muench, A. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 709
- O’Toole, S. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 413
- Quinn, P. J., Barnes, D. G., Csabai, I., et al. 2004, in *Optimizing Scientific Return for Astronomy through Information Technologies*, edited by P. J. Quinn, & A. Bridger, vol. 5493 of Proc. SPIE, 137
- Salgado, J. 2019, in *ADASS XXVIII*, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (San Francisco: ASP), vol. 523 of ASP Conf. Ser., 421
- Treloar, A. 2014, *Learned Publishing*, 27, S9. doi:10.1087/20140503
- Williamson, M. D., et al. 2016, *Scientific Data*, 3, 160018. doi:10.1038/sdata.2016.18

Author Index

- Abney, F., 175
 Accomazzi, A., 284, 353
 Adámek, K., **489**
 Afrin Badhan, M., **467**
 Aguado-Agelet, F., 417
 Akiyama, K., 143, 637
 Albert, K., **151**
 Alei, E., 597
 Alesina, F., **25**, 361, 405
 Alexov, A., 175, **209**
 Allen, A., **593**, 613, *616*, **717**, 723
 Allen, C., **155**
 Allen, M. G., 309, 445, 657, **729**
 AlSayyad, Y., 521
 Altieri, B., 49, 437
 Alvarez, R., 285
 Anderson, K., 321
 Angerhausen, D., 59
 Anglada, E., 339
 Ansdell, M., **59**
 Arabas, S., 365
 Araya, M., **75**
 Arevalo, M., 425
 Armour, W., 489
 Armstrong, R., 521
 Arnaboldi, M., 433
 Arney, G., 467
 Arviset, C., 21, 49, 285, 417, 437, 445, 459
 Asercion, J., **159**
 Baines, D., 21, 49, **249**, 445, 459
 Bakker, J., 417, 445
 Balcells, M., 317
 Ballester, P., *364*, **483**
 Barbarisi, I., 409
 Basu, A., 583
 Bauer, G., *466*
 Baumann, M., **253**
 Baume, G., 555
 Beard, S., 641
 Becciani, U., **29**
 Bellm, E. C., 485, 521
 Benatti, S., 597
 Bender, C. F., 567
 Berriman, G. B., 163, 220, **257**, 685
 Berthier, J., 459
 Bertin, E., 99
 Bertocco, S., 559
 Bhatnagar, S., 265
 Bignamini, A., 373, 597
 Blanco-Cuaresma, S., **353**
 Boch, T., 107, 253, **421**, 445, 661
 Bodewits, D., 471
 Boisson, C., **357**
 Bonaventura, N., 645
 Bonnarel, F., **313**, 329, 333, 597
 Borisov, S., 289
 Borne, K., **133**
 Bosch, J., **521**, 653
 Boussejra, M. O., **245**
 Bouy, H., 99
 Brady, P. R., 705
 Brandt, P., 265
 Brasseur, C., **397**
 Brazier, A., 705
 Briceño, C., 203
 Brogan, C., 265
 Brouty, M., 309
 Brown, A., 213
 Brown, M., **163**
 Brown, W., 629
 Bryden, G., 127
 Buchschacher, N., 25, 361, **405**
 Budavári, T., 583
 Bukovi, K., 353
 Burger, M., 175
 Burke, C., 95
 Burnier, J., 25, **361**, 405

- Bushouse, H., **543**
 Busse, D., 151

 Caballero, R., 381
 Caballero-Nieves, S. M., 83
 Cabello, C., 187, 317
 Cabot, F., 25
 Calanducci, A., 29
 Caldwell, D., 59
 Camorro-Cazorla, M., 187
 Cárdenes, R., 321
 Cardiel, N., 187, **317**
 Carrascosa, J. P. C., 151
 Carry, B., 49, 459
 Casas, F., 75
 Cáseres, R., 75
 Castillo-Morales, Á., 187
 Castro, S., 265
 Castro-Rodríguez, N., 317
 Catalán-Torrecilla, C., 187
 Ceballos, M.T., **547**
 Cenko, S. B., 705
 Chandler, C., 217
 Chang, C., 689
 Cheek, N., 339
 Chevallard, J., 645
 Chiang, H., 521
 Chilingarian, I., 33, 220, 289, **629**
 Christensen, E., 511
 Chu, S., **127**
 Chyla, R., 353
 Ciardi, D., 257
 Clarke, T., 441
 Cobo, B., 547
 Comrie, A., **17**, 265
 Conversi, L., 49
 Costa, A., 29
 Coulais, A., **365**
 Crawford, S., 697, 700
 Cresitello-Dittmar, M., 179, 429, 563
 Cui, C., 551, 729
 Curtis-Lake, E., 645

 D'Abrusco, R., 713
 Dai, C., **71**
 Damasso, M., 597
 Damian, A., 425
 Danezi, A., 677
 Davies, J., 543

 de la Calle, I., 249
 de Marchi, G., 21
 de Teodoro, P., 409, 417, 437, 445, 459
 Deelman, E., 689
 Deil, C., 357
 Delgado, A., **261**
 Delmotte, N., 433
 Deming, D., 467
 Demleitner, M., 445
 Dencheva, N., **535**
 Derriere, S., 107, 415, 657, **667**
 Desai, R., 167
 Deshpande, S., **167**
 de Souza, F. C., 103
 Diaz, R. I., **305**, 701
 Dodelson, S., 689
 Domagal-Goldman, S., 467
 Domínguez-Palmero, L., 317
 Donaldson, T., **237**, 429
 Donath, A., 357
 dos Santos, R. D. C., **103**
 dos Santos Junior, W. A., 103
 Dower, T., **369**, 380
 Dowler, P., **425**, 729
 Duchêne, G., 87
 Dullo, B. T., 187
 Durán, J. S. C., 151
 Duran, J., 417, 425, 445
 Durand, D., 277, 280, 425, 497
 Duvert, G., 365

 Ebisawa, K., **515**
 Eggl, S., 521
 Eguchi, S., 13, **493**
 Ehle, M., 503
 Eisenhower, J., 543
 Emonts, B., **265**
 Ernst, C. M., 605
 Esquej, P., 339
 Evans, D. W., 261
 Evans, J. D., 179, 377, 380, 729
 Eychaner, G., 233

 Fabbro, S., 277
 Fabricant, D., 629
 Fabrizio, G., 49
 Fan, D., **551**
 Farias, H., 579
 Feinstein, C., **555**

- Ferguson, H., **269**
Fernandez, M., 409, 437
Fernique, P., 421, **497**, 609
Ferruit, P., 645
Fiethe, B., 151
Findeisen, K., 521
Fisher-Levine, M., 521
Fitzpatrick, M., **233**
Fleming, S., 397, 453
Flinois, S., 365
Fong, W., 511
Forchí, V., 433
Ford, P., 265
Fourniol, N., 433
Fox, M., 397
Fraile, E., 339
Frailis, M., 199, 531
Fujishiro, I., 245
Fukagawa, M., 637
Fulmer, L., 233
Fulton, B. J., 257

Gabriel, C., 191, 503
Galeotta, S., 199, 531
Galkin, A., 333
Gallego, J., 187, 317
Gandorfer, A., 151
Garcia Marin, M. M., 305
Garcia, C., 381
García-Dabó, C. E., 265
Garwood, B., 265
Garzón, F., 317
Gaudet, S., 425
Geers, V. C., **641**
Gelino, C. R., 163
George, L., 167
Gianniotis, N., 709
Giardino, G., **645**
Gilda, S., **67**
Gil de Paz, A., 187
Giordano, F., 437, 459
Giordano, M., 147
Glorian, J.-M., 449
Glotfelty, K., 563
Golap, K., 265
Golkhou, V. Z., 485
González-Núñez, J., 21, 409, **417**, 445, 459
Good, J. C., 257, **685**

Gower, M., 653
Goz, D., **559**
Gracia-Abril, G., **213**
Grange, Y., 483
Grant, C. S., 353
Greene, G., 295
Greenfield, P., 535
Grishin, K. A., **33**
Guan, Y., 151
Guerra, R., **339**
Gupta, P., **171**
Gutiérrez-Sánchez, R., 417, 445
Guy, L. P., 521
Guyonnet, A., 521
Guyot, A., 107
Gwyn, S. D. J., **649**

Haber, R., 83
Hainaut, O., 433
Hale, A., 265
Hambly, N. C., 445
Hammersley, P., 317
Han, J., 551
Hanisch, R., 295
Hanson, K., 381
Hargis, J., 397, 425
Harrison, D. L., 261
Hayama, K., 493
He, B., 551
He, H., **563**
Henneken, E., 353
Hernandez, J., 21
Herrera-Fernandez, J. M., 249
Hirzberger, J., 151
Hodgkin, S., 261
Holzmann, L., 329
Honma, M., 143, 637
Hostetler, T. W., 353
Huang, L., 233
Hunter, T., 265

Ibarra, A., 249, 503
Ichinohe, Y., 79
Ikeda, S., 143, 575, 637
Indebetouw, R., 265
Iniesta, J. C. d. T., 151
Intema, H., 167
Ioannou, Y., 59
Irwin, M., v

- Isaacson, H., 257
 Ishwara-Chandra, C. H., 167
 Ivezić, Ž., 521
 Iwasaki, H., **79**

 Jakobsen, P., 645
 Jansen, F., 213
 Jarno, A., 645
 Jeeves, H., 277
 Jenkins, J. M., 59, 241, 453
 Jenness, T., 521, **653**, 701, 729
 Jeschke, E. R., 325
 Johnston, K. B., **83**
 Joliet, E., **681**
 Joncour, I., 58, **87**, 571
 Juneau, S., 233
 Jung, G., 365
 Juric, M., 401, **485**, 705

 Kahn Ahmed, M., 433
 Kaldamae, L., 381
 Kaleida, C., **175**, 209
 Kansky, J., 629
 Kaplan, K. F., **567**
 Karim, R. L., **571**
 Kassim, N. E., 441
 Kataoka, A., 637
 Katkov, I., 33, 289
 Katsavounidis, E., 705
 Kaufman, Z., **179**
 Kavak, Ü., 617
 Kavelaars, J., 277
 Kawasaki, W., 13, **37**, 265
 Kelley, M. S. P., 466
 Kelley, M. S. P., **471**
 Kempton, E. M.-R., 467
 Kent, B. R., **3**, 12, 265
 Kepley, A., 265, 587
 Khelifi, B., 357
 Kimball, A., 217
 Kirk, R. L., 605
 Kitaëff, V., **183**
 Klaassen, P. K., 641
 Kobayashi, T., 13, 37
 Kolleck, M., 151
 Kong, M., 163, 257
 Kong, X., **91**
 Kopparapu, R. K., 467
 Kosugi, G., 13, 37, 143, **575**

 Kotake, K., 493
 Kovács, G., 521
 Kretschmar, P., 503
 Krughoff, K. S., 521
 Kuik, C. L., 381
 Kulesa, C., 617
 Kurtz, M. J., 353
 Kuulkers, E., **503**
 Kyono, E., 601
 Kyprianou, M., 175, 209

 Laantee, C., 409
 Labrie, K., **321**
 Lacy, M., **217**
 Lagg, A., 151
 Laher, R. R., 471
 Laity, A. C., 163
 Lam, M., 95, **539**
 Lammers, J., **273**
 Lammers, U., 21, 213, 339, 417
 Landais, G., **309**
 Landoni, M., **373**
 Lange, T., 151
 Lange, U., 433
 Latham, D. W., 453
 Laurino, O., 179, **429**
 Lember, A., 381
 Lemson, G., 429
 Levesque, E. M., 503
 Lewis, J., 220
 Li, C., 551
 Li, J. X., 71
 Lieu, M., **49**, 58
 Liiva, H., 381
 Lim, K.-T., 653
 Lim, P. L., **325**
 Liu, W., 71
 Lockhart, K. E., 353
 Lončarić, S., 401
 Long, M., **123**
 Longmore, S. N., 95
 Loose, G., **349**
 Lopez-Caniego, M., 459
 Lorente, N., 12
 Louys, M., 313, **329**, 333, 597
 Lumi, K., 381
 Lundquist, M. J., **511**
 Luo, A.-L., 91, 119
 Lupton, R. H., 521

- Lust, N. B., 521
Lutz, K. A., **657**
- MacArthur, L., 521
Mader, J. A., 163
Maggio, G., 199, 531
Mahadevan, S., 567
Maino, D., 199, 531
Major, B., **277**, 425
Mantelet, G., 313
Maris, M., 597
Marti, B. L., 459
Martin Furones, A., 285
Martinez, B., **409**
Mascetti, L., 433
Masci, F. J., 471
Matsubayashi, K., 245
McDonald, S., 353
McEwen, A. S., 605
McGlynn, T., 220
McLaughlin, W., 179
McLean, B., 425
McWhirter, P., **95**, 539
Mechev, A. P., **677**
Mehringer, D., 265
Mellado, P., **41**
Merin, B., 21, 49, 409, 417, 437, 445, 459, 503
Meyer, J. D., 265
Meyers, J., 521
Michalik, H., 151
Michel, L., 429, **661**
Micol, A., **433**
Miel, R., 265
Miller, J., 155, 179, 377
Mink, J., **281**, 284, **701**
Mizumoto, Y., 13, 37, 621
Moellenbrock, G., 265
Molinari, S., 29
Molinaro, M., **597**
Momcheva, I., **223**, **671**
Monkewitz, S., 471
Moolekamp, F., 521
Mora, A., 21, 417, 445
Moran, S., 629
Moraux, E., 87
Morgan, E., 453
Morita, E., 13, 37
Morrison, C. B., 521
- Morton, T. D., 521
Muench, A. A., 284, 693, **713**
Mullally, S., 397, 453
Mundy, L. G., 87, 571
Muralikrishna, A., 103
Muto, T., 637
Myers, S., 217
- Nakahira, S., 515
Nakazato, T., **143**, 265, 575
Nascimbeni, V., 597
Navarro, V., **285**, 381
Nebot, A., **477**, 497, 657
Ness, J.-U., 503
Nidever, D., 233
Nieto, S., 409, **437**
Nikolic, B., **63**
Nikutta, R., 233
Ninan, J., 567
Nishie, S., 265
Noiret, D., 437
Norman, H., 459
Noumaru, J., **601**
Nullmeier, M., 313, 333
Nunez, C., 579
Nyland, K., 217
- O'Mullane, W., 521
O'Toole, S., **413**
Oberdorf, O., 425
Oberoi, D., 167
Ocvirk, P., 309
Ohishi, M., 13, 37, 621
Olsen, K., 233
Oonk, J. B. R., 677
Osborn, H. P., 59
Osinde, J., 417
Ott, J., 265
- Paegert, M., 629
Paillassa, M., **99**
Parejko, J., 521
Pascual, S., **187**, 317
Pasenkov, E., 381
Paterson, K., 511
Patrick, L. R., 317
Patterson, G. W., **605**
Patterson, M. T., 485
Paxson, C., **377**

- Pecontal, A., 645
 Peille, P., 547
 Pelló, R., 317
 Perea-Calderon, J., **191**
 Perez, H., 409
 Pérez-López, F., 285, **381**
 Perret, E., 309
 Peter, A. M., 83
 Peters, W., 441, 617
 Petit, V., 83
 Petry, D., 265
 Phillip, C., 397
 Pińska, A., 17
 Pineau, F.-X., 329, **609**
 Piqueras, L., 645
 Pizzo, R., 122, 483
 Plaat, A., 677
 Plante, R., **295**, 380
 Plazas, A. A., 521
 Pokorny, M., 265
 Polisensky, E., **441**
 Polsterer, K., 58, **709**
 Portell, J., 213
 Pound, M. W., xxi, 723
 Pouvreau, T., 309
 Pouzols, F. M., 265
 Pozo, E., 339
 Press, L., 483
 Price, P. A., 521
 Prieto, M., 317
 Primini, F., 155

 Räissi, C., 59
 Raba, R., 265
 Racero, E., 417, 437, **459**
 Ramirez, E., **21**
 Rapport, N., 353
 Ratnakumar, B., 167
 Rau, U., 265, 587
 Raugh, A., 284, **463**, 466
 Rawls, M. L., 521
 Reed, S., 521
 Renil, R., **195**
 Renshaw, A., 115
 Retzlaff, J., 433
 Reynolds, C., 265
 Richards, E. E., 441
 Riddle, R., 471
 Riebe, K., 313, 333

 Riggi, S., 29
 Riley, J., 163
 Rixon, G., 261
 Rizzi, L., 163
 Rodríguez, J., 555
 Rodríguez, J. B., 151
 Rodriguez, D. R., 425
 Rodríguez-Pascual, P., 191
 Romaniello, M., 433
 Romelli, E., **199**, 531
 Roy, A., 567
 Rubtsov, E., **289**
 Ruiz, J. E., 357
 Rusholme, B., 471, 485
 Ryan, P., **613**, 616
 Rynge, M., 689
 Ryohei, K., 637

 Sakamoto, T., 515
 Sakhadeo, A., 167
 Salazar, E., 503
 Salgado, J., 21, 249, 409, 417, 437, **445**,
 459, 503
 Salnikov, A., 653
 Sánchez-Fernández, C., 503
 Sand, D., 511
 Sanguillon, M., 313, 333, **449**
 Sasdelli, M., 59
 Saxton, R., 503
 Schaaff, A., **107**
 Schellart, P., 521, 653
 Schiebel, D., 265
 Schubert, K., 601
 Schweighart, N., 265
 Sciacca, E., 29
 Scott, A., 233
 Sealey, K., 413
 Segovia, J. C., 417, 445
 Seo, Y., 617
 Servillat, M., 313, **333**
 Shaikh, S., 167
 Shan, X., 661
 Shapurian, G., 353
 Shawhan, P. S., 704, **705**
 Shiao, B., 369
 Shibagaki, S., 493
 Shimwell, T. W., 677
 Shipman, R., **617**
 Shirasaki, Y., 13, 37, **621**

- Shortridge, K., 701
 Shupe, D. L., 471
 Shvartzvald, Y., 127
 Simmonds, R., 17
 Simpson, C., 321
 Sisodia, D., 433
 Si Lounis, A., 365
 Slater, C. T., 401, 521
 Smareglia, R., 373
 Smith, A., 223, 397
 Smith, J. C., 59, **241**
 Snyder, G., **111**
 Solanki, S. K., 151
 Solar, M., 483, **579**
 Soumagnac, M. T., 471
 Spiniello, C., 433
 Staveley-Smith, L., 183
 Stellert, M., 433
 Stephens, T., **625**
 Stoehr, F., 28, **387**, 433
 Streblyanska, A., 317
 Streicher, O., 333
 Sugimoto, K., 265
 Sullivan, I., 521
 Suoranta, V., 265
 Sutrisno, R., **115**
 Swade, D., **453**
 Swain, M. A., 163
 Swinbank, J. D., 521

 Taffoni, G., 199, 373, 531, 559
 Tafoya, D., 265
 Takekawa, S., 245
 Takeshima, Y., 245
 Tao, Y., 551
 Taranu, D., 521
 Tavagnacco, D., 199, **531**
 Taylor, A. R., 17
 Taylor, M. B., **43**
 Templeton, M. R., 353
 Tenenbaum, P., 241
 Terrien, R. C., 567
 Teuben, P. J., **xxi**, 122, 273, 613, **633**,
 723
 Teyssier, D., 213
 Thomas, B. A., **xxi**, **723**
 Thompson, D. M., 353
 Tian, F., **583**
 Timmes, F. X., **693**

 Tobar, R., 183
 Tollerud, E., **697**, 700
 Tolls, V., 617
 Tomasi, M., **147**
 Tornatore, L., 559
 Torres, S., **203**
 Tsukagoshi, T., 637
 Tsutsumi, T., 265, **587**
 Turner, J. E. H., 321
 Turtle, E. P., 605
 Twicken, J. D., 241

 Uchiki, R., 245
 Uchiyama, Y., 79
 Uemura, M., 245

 Vahi, K., **689**
 Valero-Martin, L., 249
 Vallenari, A., 213
 Vanderspek, R., 453
 Vannier, P., 309
 van Leeuwen, F., 261
 van Leeuwen, M., 261
 Varshney, A., 324
 Vastel, C., 449
 Veitch-Michaelis, J., 95
 Ventura-Traveset, J., 285
 Vera, I., 433
 Vergne, M., 555
 Vilalta, R., 115
 Vinsen, K., 183
 Vitello, F., 29
 Voutsinas, S., 445
 Vuerli, C., 199, 531

 Wadadekar, Y., 167
 Wagstaff, K. L., 127
 Walawender, J., 257
 Walker, C., 617
 Wang, J., 661
 Wang, K.-S., 265
 Wang, M., 689
 Wang, R., **119**
 Warner, E. M., **xxi**, 280, 466
 Waters, C. Z., 521
 Wells, A., 265
 Wicenec, A., 183
 Williams, P., 284
 Winegar, T., 601

744

Author Index

Witz, S., 217
Woch, J., 151
Wolf, E. T., 467
Wood-Vasey, W. M., 521
Wu, C., 183
Wu, F. Q., 71
Wu, X., 681

Xiao, J., 123
Xu, Y., 551

Yamaguchi, H., 79
Yamaguchi, M., 143, **637**
Yamanoi, H., 601
Yang, Z., 123
Ye, Q., 471
Yoldas, A., 261
Yoon, I., 265, 587
Yoshida, A., 515
Yoshino, A., 13, 37
Yu, C., 123
Yu, X. C., 71

Zampieri, S., 433
Zapart, C., **13**, 37, 621
Zečević, P., **401**
Zhang, B., 123
Zhu, M., 71, 183
Zuo, S. F., 71

Subject Index

- alerts, 261, 485, 521
- algorithm
 - alpha shapes, 571
 - CLEAN, 143, 638
 - clustering, 91
 - spatial, 87, 571
 - cross-matching, 551
 - data reduction, 493
 - DBSCAN, 87
 - deconvolution, 587
 - Delaunay Triangulation, 571
 - feature recognition, 67, 133
 - fitting, 289
 - harmonic sum, 489
 - imaging, 143, 575
 - machine learning, 95, 103, 115
 - deep learning, 49, 59, 79, 123
 - k-nearest neighbors, 67, 83
 - random forest, 111, 115
 - supervised, 67, 83, 99
 - u-net, 123
 - unsupervised, 79
 - matching, 551
 - minimization, 63
 - polygonal clipping, 567
 - RFI detection, 71
 - sky tessellation, 571
 - sparse modeling, 143
 - tiling, 649
 - Voronoi cells, 539, 571
- Amazon, cloud, 671
- applications
 - Aladin, 497
 - Aladin Lite, 261, 661
 - blender, 3
 - DS9, 269
 - Firefly, 681
 - Ginga, 325
 - Jupyter, 187, 223, 269, 317, 357, 417
 - montage, 685
 - TOPCAT, 43, 661
- archives, 167, 387, 713
 - access, 446, 449
 - design, 387
 - individual
 - ALMA Science Archive, 3, 13, 387
 - Canadian Advanced Network for Astronomy Research, 713
 - Chandra Data Archive, 713
 - Euclid Archive System, 437
 - GAIA, 43, 261, 417, 446
 - Keck Observatory Archive (KOA), 163
 - LAMOST, 119
 - LOFAR Long Term Archive, 677
 - Mikulski Archive for Space Telescopes, 713
 - NASA Exoplanet Archive, 127
 - Planetary Data System (PDS), 463, 605, 713
 - TESS, 453
 - WISE, 681
 - XMM-Newton Science Archive (XSA), 191
- JAXA, 515
- multiple
 - ESO Science Archive, 433
 - Infrared Science Archive (IRSA), 471, 681
 - Mikulski Archive for Space Telescopes (MAST), 175, 209, 223, 369, 397, 425, 453

- National Radio Astronomy Observatory (NRAO) data archive, 3
- NOAO Science Archive, 233
- astronomy
 - CMB, 147
 - comets
 - morphology, 473
 - coordinate systems, 535
 - dark energy, 199
 - dark matter, 115
 - exoplanets, 59, 127, 241, 361, 405, 453, 467, 597
 - extragalactic, 183
 - fast radio burst (FRB), 217
 - galaxies
 - evolution, 111
 - photometry, 555
 - gamma-ray, 75, 159, 357, 515, 625
 - gamma-ray bursts, 706
 - gravitational lensing, 689
 - gravitational microlensing, 127
 - gravitational waves, 493, 511, 705
 - model atmospheres
 - PHOENIX, 67
 - multi-messenger, 477, 493, 503, 705
 - neutrino, 705
 - periodic variables, 489
 - photometry
 - aperture, 155
 - point spread function (PSF), 555
 - polarization, 151
 - projections, 609
 - pulsar, 489
 - pulse detection, 489
 - radial velocity, 257, 377, 567
 - radio, 3, 123, 441, 489, 617, 638
 - interferometer, 633
 - single-dish, 633
 - redshift, 709
 - photometric, 103
 - solar, 151
 - solar system, 471
 - spectra, 91
 - spectral analysis, 33, 67
 - stars
 - seismology, 151
 - young stellar objects (YSOs), 87
 - stellar population, 555
 - time domain, 83, 471, 477, 489, 497, 503, 723
 - transients, 261, 511
 - variable stars, 83
 - X-ray, 79, 179, 563
- Astrophysics Data System (ADS), 353
- BagIt, 295
- Big Data, 49, 401, 409
- BoFs, 701, 705, 709, 713, 717
- calibration, 241, 617
 - astrometric, 213, 583, 649
 - photometric, 213, 649
 - wavelength, 317
- catalogues, 387
 - astrometric, 583
 - Gaia Data Release 2, 43, 213, 339, 446
 - Gaia Data Release 3, 417
 - individual
 - Chandra Source Catalog, 155
 - services, 661
 - VizieR, 253, 309
 - X-ray, 155
- Centre de Données astronomiques de Strasbourg (CDS), 107
- citation, 593, 713
 - management, 353, 593
- classification, 709
 - algorithms, 83, 587
- code
 - legacy, 273
 - repository
 - GitHub, 159
 - GitLab, 361
- computer languages
 - C, 163
 - csh, 163
 - Fortran, 273
 - GDL, 365
 - Go, 171
 - IDL, 163, 365
 - Java, 37, 361
 - Javascript, 33
 - Julia, 147
 - Python, 33, 63, 159, 163, 237, 273, 321, 325, 543, 575, 653, 685

Subject Index

747

Rust, 13
computers
 hardware
 ARM processors, 559
computing
 architecture, 689
 CUDA, 63
 RESTful, 203
cloud, 29, 277
 Google, 373
 Platform as a Service (PaaS),
 223
cluster
 Kubernetes, 353, 681
exascale, 559
GPU, 63, 489, 559
grid, 191
mobile, 3
 Android, 3
 iOS, 3
 iPad, 3
 iPhone, 3
resources
 Amazon Web Services (AWS),
 223, 353
virtual machines, 625
conference
 ADASS 2018, 693
configuration management, 179, 381
cross-matching, 401, 551
data
 access, 433, 449, 621
 analysis, 43, 59, 83, 103, 159, 449,
 511, 515
 processing, 485
 spectral, 289
 bad pixel masks, 563
 cube, 3, 245, 397
 hyperspectral, 579
 Hierarchical Progressive Surveys
 (HiPS), 253, 313, 421, 661,
 667
 management, 209
 archive, 209, 387
 operations, 175, 209, 339
 workflows, 245
 model, 429, 597
 CAOM, 425

CIAO, 563
multi-dimensional, 25, 79, 151
pipelines, 163, 543
 processing, 163, 191, 217, 241,
 441, 521, 677
 reduction, 183, 203, 305, 321
 science, 257, 521
preservation, 295
processing, 115, 213, 617
provenance, 333
publishing, 295
quality assurance, 309, 325
query agent, 107
reduction, 123, 587
repository, 295, 713
simulated, 241, 548
sparse, 575
spectropolarimetric, 151
storage, 387
tabular, 43
data centres
 Canadian Astronomy Data Centre
 (CADC), 277, 425
 CDS, 107, 253, 309
 ESAC Science Data Centre
 (ESDC), 213, 409, 425
 OV-GSO, 449
data formats, 295, 701
 ASDF, 269, 535, 543
 FITS, 13, 37, 543, 701
 headers, 281
 keywords, 281
 WCS, 535
 HDF5, 17
 PDS, 463
 VOTable, 429
 YAML, 543, 653
data products, 605
databases
 individual
 NASA Extragalactic Database
 (NED), 681
 VizieR, 713
 NoSQL, 405
 query language
 ADQL, 313, 369
 PostgreSQL, 441, 629
 Spark SQL, 401
 SQLite, 472

748

Subject Index

- dissemination, 657
- distribution
 - von Mises, 489
- exposure time calculator, 377
- FAIR, 295
- Flagging, 71
- Google
 - cloud, 373
- graph database, 329
- hackathon, 723
- images, 605
 - analysis, 99
 - reconstruction, 638
 - tessellation
 - HEALPix, 253, 329, 401, 423, 539, 609
 - Multi-Order Coverage (MOC), 667
 - Voronoi, 539
- instruments
 - camera
 - EIS, 605
 - configuration, 653
 - individual
 - Megacam, 649
 - MIRI, 641
 - NIRSpec, 645
 - Integral Field Spectrograph (IFS), 579
 - interferometer, 167, 265, 575, 633
 - monitoring, 41
 - polarimeter, 151
 - simulation, 641
- international collaboration, 409
- International Virtual Observatory
 - Alliance (IVOA), 237, 333, 369, 477, 597, 713, 729
 - provenance data model, 329
 - Standards, 433
- Japanese Virtual Observatory (JVO), 621
- journals, 593
 - Journal of Open Source Software (JOSS), 593
- language
 - natural, 107, 621
- libraries
 - Apache Airflow, 677
 - Astropy, 269, 535
 - astropy, 237
 - GWCS, 269, 535
 - HEALPix, 609
 - tensorfit, 579
 - WCSTools, 281
- LSST, 713
- metadata, 621
- methods
 - Bayesian, 155
 - Fourier, 489
 - Gaussian distributions, 75
 - indexing
 - R-tree, 472
 - multi-order coverage (MOC), 253, 667
 - Principal Components Analysis (PCA), 133, 579
 - robust statistics, 583
 - statistical
 - Bayesian, 583
 - Markov chain Monte Carlo, 157
 - temporal filtering, 473
- observatories
 - ground-based
 - ALMA, 3, 13, 37, 143, 575, 638
 - CFHT, 649
 - CTA, 75, 313, 357
 - FAST, 183
 - Gemini, 321
 - GMRT, 167
 - Gran Telescopio Canarias, 187, 317
 - Hobby-Eberly Telescope, 567
 - IceCube, 706
 - KAGRA, 493
 - Keck, 163
 - LIGO, 706
 - LOFAR, 677
 - LSST, 521, 653, 706
 - MeerKAT, 195
 - National Optical Astronomy Observatory (NOAO), 203

Subject Index

749

- SKA, 195
- Subaru, 601
- UKIRT, 127
- VLA, 217, 441, 629
- VLITE, 441
- NAOJ, 325
- NRAO, 3
- space-based
 - ATHENA, 547
 - Chandra, 155, 563
 - Euclid, 199, 437, 531
 - Gaia, 21, 43, 249
 - Herschel, 249
 - HST, 223, 325
 - International Space Station (ISS), 515
 - JWST, 175, 209, 223, 269, 305, 325, 543, 641, 645
 - Kepler, 59
 - SDO, 151
 - STSci, 209, 325
 - TESS, 59, 241, 257, 397, 453
 - WFIRST, 127
 - XMM-Newton, 191
- observing
 - scheduling, 503
 - time allocation, 503
- Oculus Rift, 21
- organizations
 - American Astronomical Society (AAS), 693
 - Working Group on Astronomical Software (WGAS), 693
 - ASP, 693
 - ESA, 249
 - ESAC, 381, 409
 - IRAM, 41
 - Quasar, 249
- packages
 - CIAO, 179, 563
 - Common Astronomy Software Applications (CASA), 183, 265, 587, 633
 - Docker, 250, 277, 353, 625, 681
 - GDL, 365
 - IDL, 269
 - IRAF, 281
- PRIISM, 144
- SOLR, 353, 405
- VisIVO, 29
- Vissage, 37
- physics
 - polarization, 37
 - radiation transfer, 171
- pipelines, 689
 - Transient imaging pipeline, 521
 - VDP, 441
- portals
 - ESASky, 459
- projects
 - ASTERICS, 253, 313, 597, 657, 708
 - Astronomy Data Services (ADS), 353
 - CANFAR, 277
 - CyberSKA, 17
 - Data Analysis Center for Exoplanets (DACE), 361, 405
 - NOAO Data Lab, 713
 - SCiMMA, 706
 - SciServer Compute, 713
 - StarFormMapper, 249
 - VIALACTEA, 29
- protocols
 - SAMP, 21, 250
 - TAP, 250, 369
 - VOEvent, 485, 707
- provenance metadata, 329
- radio frequency interference (RFI), 71
- registry, 621
- research
 - citizen science, 95
 - infrastructure, 613
- satellites
 - earth observation
 - MODIS, 95
- simulations, 111, 645
 - Markov Chain Monte Carlo (MCMC), 128
 - N-body, 559
- software, 163
 - applications, 613
 - calibration, 305, 543
 - containers, 277

- design, 163, 321
- development, 163, 531
 - agile, 339, 349
 - DevOps, 339
- frameworks
 - HTCondor, 175
 - Spark, 401, 437
- image analysis, 325
- image processing, 685
- infrastructure, 163
- open source, 697
- performance, 163, 531
- portability, 163
- project management, 381
- project tracking
 - JIRA, 209
- scheduling, 381
- simulation
 - MIRISim, 641
- source code, 159, 273, 613
- spectral analysis, 33, 79, 183, 289, 467
- testing, 163, 381
- tools
 - Cython, 157
 - Gtk, 325
 - Qt, 325
- user interface, 387
- user interfaces
 - web-based, 203, 493
- virtualization, 625
- workflows, 163, 245, 511
- Pegasus, 689
- space probes
 - Europa Clipper, 605
- spectrograph
 - multi-object, 317, 579, 629
- spectroscopy
 - Echelle, 567
 - longslit, 317
- stars
 - light curves, 477
- surveys
 - Catalina Sky Survey, 511
 - GAPS, 597
 - GLEAM, 413
 - LAMOST, 91
 - Mapping Nearby Galaxies at APO (MaNGA), 245
 - Pan-STARRS, 421
 - UKIRT Infrared Deep Sky Survey (UKIDSS), 33
 - VLA Sky Survey (VLASS), 217
 - Zwicky Transient Facility (ZTF), 401, 471, 485
- techniques
 - API, 203
 - convolutional neural networks (CNN), 75, 99
 - generative spectrum networks (GSN), 119
 - image deconvolution, 587
 - line fitting, 289
 - machine learning, 59, 83, 103, 241, 387, 709
 - neural networks, 63, 71, 103
 - pattern recognition, 133
 - RFI mitigation, 71, 123
 - sparse modeling, 638
- theorem
 - Brewer, 405
 - CAP, 405
 - L'Huilier, 539
 - Nyquist, 567
- triplestore, 329
- unconference, 717
- user-experience, 387
- Virtual Observatory (VO), 333, 621, 657, 667
 - individual
 - All Sky Virtual Observatory (ASVO), 413
 - Japanese Virtual Observatory (JVO), 13, 621
 - interfaces, 437, 446
- virtual reality, 21, 25
- visualization, 3, 13, 17, 325, 681, 719
 - 3D, 3, 21
 - blender, 3
 - CAVE2, 3
 - immersive-3D, 3
 - multi-dimensional, 3, 13, 25
 - Oculus Rift, 21
 - web-based, 33
- web

Subject Index751

access, 446, 613
Apache, 401
application, 261, 361, 377
development tools
 AJAX, 41
 PHP, 163
framework, 681
interface, 203
services, 397, 613, 677
technologies, 41
World Coordinate System (WCS), 535

ASCL Index

AIPS	ascl:0003.002, 245, 269, 277, 441, 477, 701
Aladin	
ascl:1112.019, 33, 253, 313, 329, 421, 433, 446, 477, 497, 551, 609, 657, 661, 667, 701	Exo-Transmit
AladinLite	ascl:1611.005, 468
ascl:1402.005, 261, 313, 387	FFTW
AMUSE	ascl:1201.015, 159, 373
ascl:1107.007, 273	Firefly
APLpy	ascl:1810.021, 477, 681
ascl:1208.017, 33	GADGET-2
AST	ascl:0003.001, 3
ascl:1404.016, 701	Gammapy
Astropy	ascl:1711.014, 357
ascl:1304.002, 3, 33, 163, 187, 203, 209, 223, 245, 249, 253, 261, 269, 277, 357, 401, 409, 417, 421, 429, 446, 535, 551, 579, 609, 657, 697	GENGA
Astroquery	ascl:1812.014, 25
ascl:1708.004, 209, 223, 253, 261, 397, 417, 657, 661, 671	Ginga
CASA	ascl:1303.020, 269, 325
ascl:1107.013, 17, 143, 183, 265, 277, 587, 633, 638	Glue
CASSIS	ascl:1402.002, 209, 269
ascl:1402.013, 449	HDS
CFITSIO	ascl:1502.009, 621
ascl:1010.001, 701	HEALPix
CIAO	ascl:1107.018, 516, 539, 609, 667
ascl:1311.006, 563	HOTPANTS
CMFGEN	ascl:1504.004, 511
ascl:1109.020, 452	Hy-Nbody, 559
DALiuGE, 183	IRAF
DIAMONDS	ascl:9911.002, 203, 223, 245, 269, 281, 543, 579
ascl:1410.001, 373	Kapteyn
DS9	ascl:1611.010, 3
	Keras
	ascl:1806.022, 71
	LEXTeS
	ascl:1711.018, 614

ASCL Index

753

McLuster	ascl:1107.016, 489
ascl:1107.015, 87	SkyMaker
MegaPipe, 649	ascl:1010.066, 99
MIRIAD	SOPHISM
ascl:1106.007, 273	ascl:1810.017, 151
Montage	spectral-cube
ascl:1010.036, 3, 273, 685	ascl:1609.017, 271
MPFIT	SPLAT
ascl:1208.019, 365	ascl:1402.007, 657
Nahoon	SPLAT-VO
ascl:1409.009, 451	ascl:1402.008, 433
NEMO	SSMM
ascl:1010.051, 273	ascl:1807.032, 83
Obit	STILTS
ascl:1307.008, 167, 441	ascl:1105.001, 43, 433, 551, 657
ParselTongue	SunPy
ascl:1208.020, 167	ascl:1401.010, 223, 697
Period04	Synspec
ascl:1407.009, 477	ascl:1109.022, 452
PHOENIX, 67	T-PHOT
ascl:1010.056, 119, 289	ascl:1609.001, 199
Photutils	TOPCAT
ascl:1609.011, 95, 269	ascl:1101.010, 43, 433, 551, 657,
PRESTO	661
ascl:1107.017, 489	Turbospectrum
PyBDSF	ascl:1205.004, 452
ascl:1502.007, 441	VisIVO
PyCBC	ascl:1011.020, 29
ascl:1805.030, 689	Vissage
PyMC	ascl:1402.001, 37, 621
ascl:1506.005, 223	WCSTools
PyRAF	ascl:1109.015, 281, 701
ascl:1207.011, 245	Xmatch
PyVO	ascl:1303.021, 551
ascl:1402.004, 433, 657	XSPEC
Saada	ascl:9910.005, 516
ascl:1111.003, 309	yt
SAOImage	ascl:1011.022, 3
ascl:0003.002, 245, 277	
scarlet	
ascl:1803.003, 521	
SEextractor	
ascl:1010.064, 387, 579	
Sherpa	
ascl:1107.005, 155	
SIGPROC	

ASTRONOMICAL SOCIETY OF THE PACIFIC



THE ASTRONOMICAL SOCIETY OF THE PACIFIC is an international, nonprofit, scientific, and educational organization. Some 120 years ago, on a chilly February evening in San Francisco, astronomers from Lick Observatory and members of the Pacific Coast Amateur Photographic Association—fresh from viewing the New Year’s Day total solar eclipse of 1889 a little to the north of the city—met to share pictures and experiences. Edward Holden, Lick’s first director, complimented the amateurs on their service to

science and proposed to continue the good fellowship through the founding of a Society “to advance the Science of Astronomy, and to diffuse information concerning it.” The Astronomical Society of the Pacific (ASP) was born.

The ASP’s purpose is to increase the understanding and appreciation of astronomy by engaging scientists, educators, enthusiasts, and the public to advance science and science literacy. The ASP has become the largest general astronomy society in the world, with members from over 70 nations.

The ASP’s professional astronomer members are a key component of the Society. Their desire to share with the public the rich rewards of their work permits the ASP to act as a bridge, explaining the mysteries of the universe. For these members, the ASP publishes the Publications of the Astronomical Society of the Pacific (PASP), a well-respected monthly scientific journal. In 1988, Dr. Harold McNamara, the PASP editor at the time, founded the ASP Conference Series at Brigham Young University. The ASP Conference Series shares recent developments in astronomy and astrophysics with the professional astronomy community.

To learn how to join the ASP or to make a donation, please visit <http://www.astrosociety.org>.

ASTRONOMICAL SOCIETY OF THE PACIFIC MONOGRAPH SERIES

Published by the Astronomical Society of the Pacific

The ASP Monograph series was established in 1995 to publish select reference titles.

For electronic versions of ASP Monographs, please see

<http://www.aspmonographs.org>.

INFRARED ATLAS OF THE ARCTURUS SPECTRUM, 0.9-5.3 μ m

eds. Kenneth Hinkle, Lloyd Wallace, and William Livingston (1995)

ISBN: 1-886733-04-X, e-book ISBN: 978-1-58381-687-5

VISIBLE AND NEAR INFRARED ATLAS OF THE ARCTURUS SPECTRUM 3727-9300Å

eds. Kenneth Hinkle, Lloyd Wallace, Jeff Valenti, and Dianne Harmer (2000)

ISBN: 1-58381-037-4, e-book ISBN: 978-1-58381-688-2

ULTRAVIOLET ATLAS OF THE ARCTURUS SPECTRUM 1150-3800Å

eds. Kenneth Hinkle, Lloyd Wallace, Jeff Valenti, and Thomas Ayres (2005)

ISBN: 1-58381-204-0, e-book ISBN: 978-1-58381-689-9

HANDBOOK OF STAR FORMING REGIONS: VOLUME I THE NORTHERN SKY

ed. Bo Reipurth (2008)

ISBN: 978-1-58381-670-7, e-book ISBN: 978-1-58381-677-6

HANDBOOK OF STAR FORMING REGIONS: VOLUME II THE SOUTHERN SKY

ed. Bo Reipurth (2008)

ISBN: 978-1-58381-671-4, e-book ISBN: 978-1-58381-678-3

TWENTY YEARS OF ADASS

ed. Ian Evans (2013)

ISBN: 978-1-58381-822-0, e-book ISBN: 978-1-58381-823-7

A complete list and electronic versions of ASPCS volumes may be found at

<http://www.aspbooks.org>.

All book orders or inquiries concerning the ASP Conference Series, ASP Monographs, or International Astronomical Union Volumes published by the ASP should be directed to:

Astronomical Society of the Pacific

390 Ashton Avenue, San Francisco, CA 94112-1722 USA

Phone: 800-335-2624 (within the USA)

Phone: 415-337-2126 Fax: 415-337-5205

Email: service@astrosociety.org

For a complete list of ASP publications, please visit <http://www.astrosociety.org>.

