NFEST/2018/R–158

An Overview of the Astronomical Data Processing Software Systems in Twenty First Century

Yasha Kaushal[1], Vijay Kumar Doddamani[2], Sameer Patel[3]

[1]*NIT-Bhopal, Bhopal- 462003, India*
[2]*Department of Physics, Bangalore University, Bangalore-560059, India*
[3]*Charotar University of Science and Technology, Gujarat, Anand-388421, India*
[1]*yasha.kaushal333@gmail.com*

Abstract: *Data processing and analysis techniques in astronomy have grown in sophistication over the past few decades owing to the advancements in computational techniques and big data processing. Here we present a brief overview of several rapidly advancing and emerging software systems employed in astronomical data reduction and processing. We will elaborate upon their recent developments, advantages, limitations, applications and future potentials.*

Keywords: *AIPS; AIPS++; CASA; Astropy; CIAO; ESO-MIDAS; IRAF; PyRAF; HEASoft; XANADU; FTOOLS.*

1. INTRODUCTION

Astronomical observations produce stupendous amount of data. For current and future planned instruments to routinely meet their performance goals, standard analysis techniques must be significantly improved, leading to at the same time more sophisticated structure. It is important that programming environments for astronomical software permit fast and flexible development. Recent advances in astronomical observing and simulation facilities are expected to move astronomy towards a new data-intensive era of 'big data'. They enable the astronomers to explore the Universe at a spatial and frequency resolution which was never done before but handling the data from these facilities which is roughly of terabyte to petabyte order per observation, will pose significant challenges for the present astronomical data analysis and synthesis tools [1].Three major challenges for next few decades are: (1) CPU processing rates are leveling off with rapidly growing data rates; (2) computationally quick and efficient analysis techniques are becoming integral part of astronomy; and (3) industrial advancements in hardware and software computing influence major changes in astronomical algorithms [2].

Astronomical data analysis systems in use today include AIPS, AIPS++ and CASA for radio astronomy; CIAO and HEASOFT for X-ray astronomy; and IRAF/PyRAF, MIDAS and MOPEX for optical and infrared astronomy. They all represent several million lines of code and years of development effort. Table 1 classifies the software systems on the basis of their primary operation wavelength. Individual researchers and Astronomy departments across the world rely heavily on these systems to reduce and analyze the data.

There are some issues with few software systems as well like, limited or no access to the source code andrelatively high license fees (as with IDL and MATLAB). Nonetheless, smaller teams are developed for the evolution and maintenance of such software libraries and simultaneously leading to the trend of open-source scientific and statistical packages such as R, Astropy and PyRAF.

TABLE 1

S.No	Software System	Wavelength of Operation
1.	AIPS	Radio*
2.	AIPS++	Radio*
3.	CASA	Radio
4.	Astropy	Optical, Radio,
5.	CIAO	X-ray
6.	ESO-MIDAS	Optical, Infrared
7.	IRAF/PyRAF	Optical, Infrared **
8.	HEASoft	X-ray, Gamma-ray

**Some routines available for processing of image data at Infrared, Optical, Ultraviolet and X-ray.*
***External packages supported for EUVE (Extreme Ultra-Violet) and Chandra (X-ray).*

2. DESCRIPTION OF SOFTWARE SYSTEMS

2.1 AIPS

Astronomical Image Processing System (AIPS) is a software package developed and maintained by NRAO(National Research and Astronomical Observatory) and meant for the interactive calibration and editing of radio interferometric data and for the display, analysis and calibration of astronomical images retrieved from those data using Fourier analysis. The new development version of AIPS ready for installation is **31DEC17**. AIPS is written in FORTRAN and C and was release under the Free Software Foundation's General Public License in mid-1995. The AIPS package has capabilities of calibration and editing of functions for both VLBI (Very Long Baseline Interferometry) and VLA (Very Long Array) data with interactive and batch modes [3]. VLA is an aperture synthesis instrument with 27 individual elements and capable of producing enormous quantities of data. For instance, every ten seconds, a spectral-line observation can produce about 90000 measurements [4]. AIPS can also be used to work on single dish measurements [5].

AIPS can perform global fringe-fitting, geometric corrections in addition to the standard calibrations for connected-element interferometers, and offers special phase-referencing and calibration for VLA. Since 2010, considerable efforts have been made to update AIPS to deal with the wide bandwidths and other improvements of the Expanded VLA which is now known as Jansky VLA.

AIPS is a powerful tool to display and analyze two- and three-dimensional radio images from the NRAO's Very Large Arraysince early 1980s. It created foundation for self-calibration and imaging of VLA continuum and spectral-line data. It contains facilities for image construction by Fourier inversion; for displaying and editing of data in the aperture or u-v plane; for filtering, image combination and parameter estimation; and for a wide variety of image and graphical displays [6].

AIPS uses a command language which is used to run "tasks" and to interact with text, graphics and image displays, along with a batch mode support. Other astronomical applications of AIPS include the processing of image data at infrared, visible, ultraviolet and X-ray wavelengths and the display and analysis of line and continuum data from large single-dish radio surveys.POPS is command line interpreter of AIPS. ParselTongue makes AIPS algorithms and data structures available in Python. It is suitable as a scripting interface for doing data reduction on large data sets. It is also used as a coding platform for the new calibration algorithms that are being developed as part of the ALBUS (Advance Long Baseline User Software) project. ParselTongue also supports remote execution of tasks using Python's built-in XML-RPC that can run tasks in parallel on a cluster. Developments are going on for a work-flow manager to create and control ParselTongue scripts. ParselTongue has a potential to form the basis for a Graphical User Interface (GUI) to AIPS [7].

The AIPS++ and now CASA replace all of the functionality of AIPS with additional functionalities.

2.2. AIPS++

AIPS++ is the next stage evolution of AIPS. It provides facilities for calibration, image formation, image enhancement and analysis. Though the packages in *AIPS++* mainly process the data from varieties of radio telescopes, they are also useful for processing other types of astronomical data as well [8]. AIPS++ now not only replaces predecessor packages but also handle new problems like non isoplanatic imaging and fitting models to the three dimensional spectral line cubes [9].

AIPS++ has been designed to be tool-based and potentially programmable by the user with tasks written in **C++** for efficiency. Several C++ classes developed for AIPS++ project are used in operations like indexing, enquiry, iteration. These classes handle arrays having an arbitrary number of dimensions. [11]. It represents object oriented approach to programming i.e., encapsulation, inheritance and polymorphism (which FORTRAN does not support). The counterpart of an AIPS task is an AIPS++ tool function. In *AIPS++*, a tool is a collection of related functions (tool functions) that operate on some common data source and constructed on demand. The counterparts of AIPS adverbs are the parameters of AIPS++tool functions. The command-line interpreter (CLI) in AIPS is POPS, while the counterpart in AIPS++ is Glish. It is an interactive programming language that allows high level operations on vectors and arrays and supports Automatic Variable Typing. All capabilities of interest to the user are controllable from this command line window- all data accessed and modified from *Glish* can be placed back into *AIPS++*.

AIPS++ code base was upgraded to a more modular structure on August 25, 2004 and renamed as CASA (Common Astronomy Software Applications). It is now the basis of the image processing systems for several next-generation radio telescopes including ALMA (Atacama Lyarge Millimeter Array), eVLA, and ASKAP [10]. The AIPS++ data system is based on "tables", which can be transported to different file formats. It allows uniform interface to all data files. Each column or row in a table can be treated as a normal vector. An X-based browser allows a user to view and edit all data items in the table. SQL (Selection Query Language) allows for selection and sorting for rows and columns [12].

A typical AIPS++ data reduction session will have Data Loading, Data Examination, Calibration, Imaging, Self Calibration, Image Analysis, Image Display and Contour plots. In Generic Processing one will have Synthesis Calibration, Imaging, Deconvolution and Self-calibration, Wide field imaging, Mosaicing (Multi-field imaging) and GBT (Green Bank Telescope) Continuum Single dish Imaging and Analysis. In Telescope specific processing one will have data reductions for VLA, GBT, ATCA (Australia telescope compact array), JCMT (James Clerk Maxwell Telescope), BIMA (Berkeley Illinois Maryland Association) and Arecibo Observatory Spectral Line Reduction [9].

2.3. CASA

CASA (Common Astronomy Software Applications) is a comprehensive software package to calibrate, image, and analyze radio astronomical data from interferometers such as ALMA and VLA and from some single dish telescopes as well. CASA's latest version is 5.0.0, released on 18 July 2017. It has a Python command line interface and table-based data model that is capable of handling large observational data with facilities for loading and storing data into the database from external formats. Since Glish command line interpreter is replaced by Python bindings, the system is now also known as "CASApy" [15]. CASA contains higher-level functions that encapsulate various astronomically important algorithms like image deconvolution and image formation. Proper documentation is available for all aspects of the packages [13].

CASA includes high-level Python interfaces to lower-level routines manifested in C++ and FORTRAN. It has a suite of C++ libraries derived from AIPS++ tasks. One can easily import and use any external Python library into the CASA environment. SciPy and PyLab package libraries are distributed along with CASA [14]. The ALMA Pipeline is being developed as a set of additional tasks based on the CASA data reduction package with implementation of end-to-end processing for interferometry and single dish data. The ALMA Pipeline aims at providing standardized data products ready for scientific analysis by automatically reducing all data taken in standard observing mode [16]. CASA's imager can do simultaneous multiscale and multifrequency deconvolution (Rau & Cornwell 2011). One remarkable point is that the CASA bundles its own Python interpreter, rather than linking to one provided by the operating system [17]. The command line interface has two levels: **Tools** which are Python objects that can access all functionality of the CASA libraries. Detailed documentation is available for the methods of these tools through the help command. And other is, **Tasks**, which areParameterized built-in Python scripts for the selected common analysis procedures. These scripts are robust and user friendly. A number of shared global parameters can be stored and recalled separately for each task and automatically checked for correctness before task execution. Users can manipulate the tasks with their own scripts [19].

The eXtended CASA Line Analysis Software Suite (XCLASS) is a toolkit for CASA, containing new functions for modeling single dish and interferometric data. For example, 'myXCLASS' tool calculates synthetic spectra by solving the radiative transfer equation for an isothermal object in one dimension with consideration of the finite source size and dust attenuation [18].The toolbox also provides an interface to the model optimizer package MAGIX (Modeling and Analysis Generic Interface for eXternal numerical codes), which helps to find the best description or fit of observational data using myXCLASS.

Typical steps involved in ALMA data analysis are Import, Flagging, Calibration, Imaging and Self calibration and Image analysis. Importing allow users to import ALMA and EVLA/VLA data as well as UVFITS. **Flagging** removes thebad data. **Calibrating** makes the measured quantities near to the ideal measurements for interferometric data. **Imaging** involves image based deconvolution and stages of Fourier transforming observed interferometric data. Various weighting schemes are employed (like natural, uniform, super uniform and Briggs) before Fourier analysis. The resulting dirty images are then deconvolved via multiple deconvolution algorithms.For **Image analysis**, CASA uses the PyLab Python library. The plotter plots simple tables, calibration data and measurement sets. CASA viewer can handle a large number of frequency channels at a time. [14]

2.4. Astropy

Astropy is an open-source and community-developed Python package containing classes and functions for interactive data analysis. It provides support for Virtual Observatory (VO) tables, ASCII table formats, unit and physical quantity conversions, celestial coordinate and time transformations, world coordinate system (WCS), a framework for cosmological transformations and conversions and for domain-specific file formats such as flexible image transport system (FITS). It is under active development for features such as a model fitting framework, VO client and server tools, and aperture and point spread function (PSF) photometry tools [20]. Its release cycle is biannual with LTS (Long term support) which guarantees a stable API release support for 2 years. It has 5 major public releases (first release February 2013) with latest stable version v1.1.2 released on 10th March 2016 [21]. Astropy aims to minimize the effort of handling data in different file formats. It achieves so via a registry of I/O "connectors" that allow such files to be transparently read into and serialized from Astropy's generic NDData and Table classes [22]. Diversity in functionality is attained by affiliated packages. An affiliated package is a python package that is not part of the core astropy package, but has requested to be included under the umbrella of the Astropy Project.

Astropy is a data mining toolkit for exploring large data cubes in radio astronomy from facilities like ALMA or CARMA (Combined Array for Research in Millimeter-wave Astronomy). It can perform UVES (Ultraviolet and Visual Echelle Spectrograph) Spectroscopy analysis, make a plot with both redshift and universe age axes, read and plot catalog information from a text file, and use its coordinates to match catalogs and plan observations t can also analyze the optical afterglow of gamma-ray bursts [23]. The High Energy Astrophysics Science Archive Research Center (HEASARC) refers to Astropy as "A single core package for Astronomy in Python". On the similar footsteps, Sunpy is also getting developed for analyzing solar and heliospheric datasets in Python. Gammapy is also community-developed, open-source Python package for gamma-ray astronomy [24].

2.5. CIAO (Chandra Interactive Analysis of Observations)

The commendable science capabilities of the Chandra X-ray Observatory (CXO) demanded new, flexible and multi-dimensional software to analyze the data it returned. This need gave rise to CIAO. It useful for the analysis of data from non-X-ray missions as well, thereby emerging as a powerful mission independent software system. CXO data is 4-dimensional -2 spatial, 1 time, 1energy) and each dimension has many independent elements. CIAO is built to handle N-dimensional data without concern about which particular axes is being analyzed.

Its latest release includes CIAO Scripts package 4.9.4 and CALDB 4.7.5.1 (20 Jul 2017). CALDB is the directory and indexing structure that provides access to all calibration files that are required for conventional processing and analysis. The CIAO features are comparable to those typically found in expensive commercial packages like IDL and MATLAB [25]."Data subspace" will keep track of how the data is being filtered and binned. The tools will maintain a log of this subspace and allow users to review previous data processing. CIAO can extract spectrum for a point-like source, extract high-resolution grating spectrum, combine spectra, convert count rate to flux, estimate the flux for a point-like source, make exposure corrected images, create 3-color (RGB) image, compute source centroid, extract light curve (with check for variability), remove background flares, register (align) fields, create mosaics and smooth image for publication.

Sherpa is the modeling and fitting application of CIAO and performs forward fitting of models to data in N dimensions. It enables the user to construct complex models and fit those models to data using a various statistical methods. *Sherpa* for CIAO v.4.9 was released on December 16, 2016 and is supported by Python 2.7 and Python 3.5. S-Lang users can also access the high-level user-interface functions to perform fits with the PySL package. Sherpa can be used to analyze images and spectra from a variety of sources like ROSAT (Röntgen Satellite) and Hubble. Sherpa has interoperability with other packages like PyFITS, DS9 and XSPEC owing to new modular design and integration with Python. [26]. With Sherpa one can fit 1-D data sets (in batch or individually) like spectra, surface brightness profiles, light curves, general ASCII arrays. It can also performfitting of 2-D images/surfaces in the Poisson/Gaussian regime.

ChIPS (Chandra Imaging and Plotting System) is the imaging and plotting platform for CIAO. It provides high-quality plots from both interactive and scriptable interfaces. It is used to plot a light curve and to create publication-quality figures. The CIAO v.4.9 release of ChIPS provides support for Python 3.5. Python users can now import ChIPS as a module that provides access to all ChIPS plotting functions. This importing provides scope for powerful new script and improvement of popular CIAO contributed scripts [27].

2.6. ESO-MIDAS

The ESO-MIDAS (European Southern Observatory-Munich Image Data Analysis System) is a software system for data reduction and image processing with special reduction packages for ESO instrumentation at La Silla and the VLT at Paranal. It contains application packages for stellar and surface photometry, image sharpening, image decomposition, and various others. ESO-MIDAS is available under the General Public License. In the design of MIDAS, a number of factors were taken in account to ensure evolution of the system in future. Some of them are modular design, portability, standard programming languages (FORTRAN and C) and standard system (X Window) [28].

MIDAS uses MIDAS command language (MCL) which provides the tools to construct complex command procedures from existing commands and also provides the basic features of a programming language like conditional statements and branching, definition of parameters, looping, procedure calls with parameters, and built-in functions.MIDAS is based on three sets of general interfaces, namely: a) “Standard Interfaces” for general I/O and image access; b) “Table Interfaces” for access to table structures; c) “Graphics Interfaces” for easy inclusion of graphical representation of the MIDAS data structures.Data items in the MIDAS system can be divided into several groups: Frames, Tables, Descriptors, Keywords and Catalogues. Applications perform the actual operations on the data. They are split in hierarchy of importance. Top level comprises of core applications integral to image processing- Image Display, General Image Processing, Graphics Display, Table File System, Fitting Package and Data I/O.

Finland joined the European Southern Observatory in 2004, providing a contribution of a significant resource called the Sampo project. The Sampo project ran for three years and prepared the ESO community for the data analysis and reduction challenges of the next decades. The first major Sampo project was PyMidas, an interface from Python to the ESO-MIDAS data analysis and reduction system. [29].In 2010s, automated data reduction workflows for astronomy gave rise to “The ESO Reflex environment”. It drastically improved the efficiency of data reduction by using automatic workflows to execute a sequence of data reduction steps. It allows execution and modification of the data reductionworkflow with facilities for inspection and interaction with the data [30].

2.7. IRAF/PyRAF

IRAF (Image Reduction and Analysis Facility) system is employed for optical and infrared data processing. It provides a large number of basic tools CCD data reductions. It can now also reduce Hubble Space Telescope (STSDAS) and X-ray astronomy (PROS) data. As a result of the combined efforts put in 1990s towards user interface, high level languages, networking, object oriented software structure and methodology and data base technology, IRAF has evolved with tremendous potential [31]. IRAF's latest version is V2.16, updated on March 22, 2012, featuring Virtual Observatory (VO) capabilities. STScI (Space Telescope Science Institute) has released PyRAF-Python scripting of IRAF tasks- a great advantage for batch processingdata.

IRAF distribution includes general image processing and graphics programs, plus a large number of packages for the reduction and analysis of optical and infrared astronomical data (e.g. "noao"). External packages are available for retrieving and handling data from other observatories/wavelength regimes such as the Hubble Space Telescope (optical), EUVE (extreme ultra-violet) or Chandra (X-ray). IRAF can process stellar Long slit, Echelle and Hydra (multifiber) spectroscopy. Basic spectroscopic reduction operations includes 1)For fibers-extract, flat field, wavelength calibrate, sky subtract; 2)For slits-extract and sky subtract, wavelength calibrate and flux calibrate; 3) For Scanners- Flat field, wavelength calibrate, beam switch and flux calibrate. Generic spectroscopy packages are 1) apextract-aperture spectra extraction package; 2) longslit- longslit reduction package; 3) onedspec- one dimensional spectral reduction and analysis package; 4) rv- radial velocity analysis package [32].

IRAF has a programmable Command Language scripting facility but has no error or exception handling. If an error occurs, the script halts immediately with an error message insufficient to trace the root cause and making it difficult to debug scripts. Python interface to IRAF tasks-called PyRAF- provides more tools than the existing IRAF CL and serves as a strong environment for programming. PyRAF is simply a command language for running IRAF tasks that is based on the Python scripting language. It gives users the ability to run IRAF tasks in an environment that has all the power and flexibility of Python. PyRAF can be installed along with an existing IRAF installation from 'stsci_python' package; one can then choose to run either PyRAF or the IRAF CL. The current release of PyRAF is v2.1.14. One can access all the features of Python including built-in list (array), dictionary (hash table) data structures, extensive string handling and GUI libraries that make it straightforward to wrap IRAF tasks with GUI interfaces. One can read data arrays and tables directly into Python and manipulate them on the command line [33].

Graphics kernel for PyRAF supports multiple graphics window, a scrollable message input/output region, ability to recall previous plots and automatic focus handling [34]. The availability of Python interfaces to systems like Midas-PyMidas, AIPS-Parseltongue, ALMA-PySL will make it easier to write scripts and combine capabilities from different systems [35].

2.8. HEASoft (XANADU and FTOOLS)

X-ray astronomy is quite challenging since pointing of the telescope is not kept constant during observations and the imaging point spread function (PSF) and spectral energy response requires precise calibration as they vary greatly across the field of view. All the X-ray data have low count rate and the cosmic rays require careful statistical attention. Despite these complexities, packages like CIAO and HEASOFT are mature enough to allow non-specialist users to reliably extract data from X-ray observations. [36]. HEASoft (High Energy Astrophysics Software) is a unified release of FTOOLS and XANADU software packages. Latest version of HEASoft is v.6.21 released on April 5, 2017. XANADU (data analysis for x ray astronomy) contains three high-level, multi-mission tasks for X-ray astronomical data analysis- Xspec (for spectral), Xrono (for timing) and Ximage (for imaging).

XSPEC 12.9.1 performs spectral fitting using the observed spectrum, the instrumental response and the model spectrum which is calculated within XSPEC using the energy ranges defined by the response file. XSPEC is designed to support multiple input data formats but support for the earlier SF and Einstein FITS formats are removed. HEASoft is working over the support for ASCII data which will allow XSPEC to analyze spectra from other wavelength regions (optical, radio) transparently to the user.

XRONOS v.5.21 is designed mostly for X-ray astronomy. XRONOS is basically a detector and wavelength-independent instrument. It has been used to analyze data from optical photometry and observatories like the Einstein Observatory. It contains packages for light curves, hardness ratio and colour-colour plotting, epoch folding, power spectrum, autocorrelation, cross-correlation, time skewness and statistical analysis. All the tasks use the Xanadu Paramater Interface based on the IRAF parameter file style. This is the standard interface used by all tools distributed within HEAsoft. There are a number of useful programs, distributed with HEASoft that manipulate the XRONOS parameter files like 'plist', 'pget', 'pset' and 'punlearn'.

XIMAGE v4.4 is a multi-mission instrument independent X-ray image display and analysis program. It supports theanalysis of data from any X-ray imaging detector if proper calibration files are available. Basic analysis of optical, infrared and radio images is also possible. One can perform image rebinning, image smoothing, source detection, statistical analysis, correction for background, exposure and point spread function, image mosaics, contour plots, sky grids and pixel-coordinate conversions, equinox changes, x/y image slices and extraction of spectra and light curves from event data.

The Swift analysis software is part of HEASoft system used to analyze Gamma-Ray Bursts (GRBs) and their X-ray and optical afterglows measured to arc-second accuracy within a minute or two by NASA's Swift satellite until they fade from view for up to months later. Swift has been finding around 90 new bursts a year since launch in November 2004. It has given the most complete study of GRBs so far and has found the most distant objects in the Universe. Swift is ideal for observations of transient and variable sources.

3. CONCLUSION

We have thrown light upon the basic functionality, applications, deployment, on-going innovations, advantages and disadvantages corresponding to each software system discussed in the paper. Since past two decades a general trend of increasing flexible programming language interfaces like Python is witnessed. Torrents of data are posing a real challenge for astronomical computation. Parallely, there is a future scope for up to the pace development of these software systems as well.

Software systems in astronomy begin with a few private efforts, but eventually become useful for the entire community only when standardization of the techniques takes place and distributed with proper documentation and interfaces. This minimizes the efforts to learn and apply the technique to future data reduction and analysis. The ultimate goal is not simply to gather data, but to create knowledge. Hence it is important to consider and work upon the process of converting data to knowledge while taking care of the potential improvements for the coming generation. Keeping in mind the limited resources we have, we need to work more closely together across agencies, projects, institutions and disciplines to give rise to the best techniques of the time for astronomical data processing and thereby avoiding redundant efforts across globe.

ACKNOWLEDGEMENT

We are thankful to the faculties and researchers in Indian Institute of Astrophysics (IIA), Bangalore and Bangalore University (BU), Bangalore for sharing their experiences with the software systems discussed in this paper. We would also like to acknowledge Bangalore University forproviding the resources and web access to the esteemed international journal publications that helped us to do a better research.

REFERENCES

[1] A. H. Hassan, C. J. Fluke, D. G. Barnes and V. A. Kilborn, "Tera-scale astronomical data analysis and visualization", Monthly Notices of Royal Astronomical Society, November 2013.

[2] Henry C. Ferguson, Space Telescope Science Institute

[3] https:// www.aips.nrao.edu

[4] Eric W. Greisen, "The Astronomical Image Processing System", AIPS Memo no. 61, National Radio Astronomy Observatory September 12, 1988.

[5] Alan H. Bridle, Eric W. Greisen, "The NRAO AIPS Project — A Summary", *AIPS* Memo 87, National Radio Astronomy Observatory, April 5, 1994.

[6] http:// www.aips.nrao.edu

[7] Mark Kettenis, Huib Jan van Langevelde, Cormac Reynolds(Joint Institute for VLBI in Europe), Bill Cotton(NRAO), "ParselTongue: AIPS Talking Python", Astronomical Data Analysis Software and Systems XV, ASP Conference Series, Vol. 351, 2006

[8] B. E. Glendenning (NRAO), " Creating an Object Oriented Software System- The AIPS++ Experience", Astronomical Data Analysis Software and Systems V, ASP Conference Series, Vol. 101, 1996

[9] G. A. Croes(NRAO), "On AIPS++, A New Astronomical Information Processing System", Astronomical Data Analysis Software and Systems II, ASP Conference Series, Vol. 52, 1993

[10] http:// en.wikipedia.org

[11] A. *G.* Willis, M.P. Healey (DRAO), B.E. Glendenning (NRAO), "AlPS+ + N-Dimensional Array Classes", Scientific Programming, Vol. 2, pp. 239-246 (1993), CCC 1058-9244/94/040239-08.

[12] G. van Diepen(NFRA), "AIPS++ Table Data System", Astronomical Data Analysis Software and Systems III, ASP Conference Series, Vol. 64, 1994

[13] J. P. McMullin, B. Waters, D. Schiebel, W. Young, K. Golap (NRAO), "CASA Architecture and Applications", Astronomical Data Analysis Software and Systems XVI, ASP Conference Series, Vol. 376, 2007

[14] S. Jaeger (Department of Physics & Astronomy, University of Calgary), "The Common Astronomy Software Application (CASA)", Astronomical Data Analysis Software and Systems XVII, ASP Conference Series, Vol. 394, 2008.

[15] https://en.wikipedia.org/wiki/CASA

[16] Dirk Muders, Friedrich Wyrowski, John Lightfoot, Stewart Williams, Takeshi Nakazato, George Kosugi, Lindsey Davis, and Jeff Kern, "The ALMA Pipeline", Astronomical Data Analysis Software and Systems XXIII, ASP Conference Series, Vol. 485, 2014 Astronomical Society of the Pacific.

[17] PETER K. G. WILLIAMS, CASEY J. LAW, AND GEOFFREY C. BOWER (Department of Astronomy, University of California, Berkeley), "Rapid Development of Interferometric Software Using MIRIAD and Python", PUBLICATIONS OF THE ASTRONOMICAL SOCIETY OF THE PACIFIC, 124:624–636, 2012 June 2012.

[18] T. M¨oller, C. Endres, and P. Schilke, "eXtended CASA Line Analysis Software Suite (XCLASS)",

Astronomy & Astrophysics manuscript no. XCLASS˙paper˙˙2-column, ESO, October 24, 2016

[19] Dirk Petry (ALMA Regional Centre, ESO), "Analysing ALMA data with CASA", ADASS XXI, ASP Conference Series, 17 Jan 2012 Astronomical Society of the Pacific.

[20] The Astropy Collaboration, "Astropy: A community Python package for astronomy", Astronomy and Astrophysics, 558, A33 (2013), DOI: 10.1051/0004-6361/201322068, ESO.

[21] Brigitta Sipőcz, "The Astropy Project", http://www.astropy.org.

[22] Erik M. Bray (STScI), "Astropy: Building Blocks for Astronomy Software", Astronomical Data Analysis Software and Systems XXIII, ASP Conference Series, Vol. 485, 2014 Astronomical Society of the Pacific.

[23] https://en.wikipedia.org/wiki/Astropy

[24] Stuart Mumford, David Pérez-Suárez, Steven Christe, Florian Mayer, Russell J. Hewettk, "SunPy: Python for Solar Physicists", PROC. OF THE 12th PYTHON IN SCIENCE CONF. (SCIPY 2013).

[25] Michael S. Noble, "Advances in CIAO Interactive Analysis", Chandra X-Ray Centre Newsletter, March 2001.

[26] John Houck, Michael Nowak (MIT), Brian Refsdal, Gregg Germain, Douglas Burke, Stephen Doe, Ian Evans, Janet Evans, Jonathan McDowell, Warren McLaughlin, Robert, Milaszewski, Joseph Miller, Dan Nguyen, Christopher Stawarz, Aneta, Siemiginowska, "CIAO 4 Preview: Sherpa and ChIPS Demo", Astronomical Data Analysis Software and Systems XVI, ASP Conference Series, Vol. 376, 2007.

[27] Elizabeth C. Galle, Craig S. Anderson, Nina R. Bonaventura, D. J. Burke, Antonella Fruscione, Nicholas P. Lee, and Jonathan C. McDowell, "Python Scripting for CIAO Data Analysis", Astronomical Data Analysis Software and Systems XX, ASP Conference Series, Vol. 442, 2011 Astronomical Society of the Pacific.

[28] Reign H. Warmels, "The ESO-MIDAS System", poster presentation in Astronomical Society of the Pacific Conference, 1992.

[29] Richard N. Hook, Sami Maisala, Tero Oittinen, Marko Ullgren, Kari Vasko, Ville Savolainen, Johan Lindroos, Matti Anttila, Otto Solin, Palle Møller, Klaus Banse, Michele Peron, "PyMidas–A Python Interface to ESO-MIDAS", Astronomical Data Analysis Software and Systems XV, ASP Conference Series, Vol. 351, 2006.

[30] W. Freudling, M. Romaniello, D. M. Bramich, P. Ballester, V. Forchi, C. E. García-Dabló, S. Moehler, and M. J. Neeser, "Automated data reduction workflows for astronomy- The ESO Reflex environment", Journal of Astronomy and Astrophysics, 559, A96 (2013), DOI: 10.1051/0004-6361/201322494.

[31] Doug Tody (NOAO), "IRAF in the Nineties", Astronomical Data Analysis Software and Systems II, Astronomical Society of Pacific Conference Series, Vol. 52, 1993.

[32] F. Valdes (IRAF Group, NOAO), "The IRAF Spectroscopy Reduction Packages and Taska", Astronomical Society of Pacific Conference Series, Vol. 25 1992

[33] P. Greenfield, R.L. White (STScI), "A New CL for IRAF Based on Python", Astronomical Data Analysis Software and Systems IX, ASP Conference Series, Vol. 216, 2000, Astronomical Society of the Pacific.

[34] M.D. De La Pena, R. L. White, P. Greenfield, "The Pyraf Graphics System", Astronomical Data Analysis Software and Systems X, Astronomical Society of Pacific Conference Series, Vol. 238, 2001.

[35] P. Greenfield, R.L. White (STScI), "Where will Pyraf Lead Us? The Future of Data Analysis Software at STScI", The 2005 HCT Calibration Workshop, STScI.

[36] Jonathan C. McDowell, "X-ray Astronomy and the Analysis of X-ray Data", Astronomical Data Analysis Software and Systems XV, ASP Conference Series, Vol. 351, 2006

[37] D. S. Berry, K. Reinhold, T. Jenness, F. Economou, "CUPID: A Clump Identification and Analysis Package", Astronomical Data Analysis Software and Systems XVI, ASP Conference Series, Vol. 376, 2007.