The ChiVO Library: advanced computational methods for astronomy

Mauricio Araya[1], Mauricio Solar[1], Diego Mardones[2], Luis Arevalo[1], Marcelo Mendoza[1],Camilo Valenzuela[1], Teodoro Hochfärber[1], Martín Villanueva[1], Marcelo Jara [1], and Axel Simonsen[1]

[1]*Universidad Técnica Federico Santa María, Valparaíso, chile;* `name.surname@usm.cl`

[2]*Universidad de Chile, Santiago, Chile;* `dmardones@uchile.cl`

Abstract. The main objective of the Advanced Computational Astronomy Library (ACALib) is to ensemble a coherent software package with the research on computational methods for astronomy performed by the first phase of the Chilean Virtual Observatory between years 2013 and 2015. During this period, researchers and students developed functional prototypes, implementing state of the art computational methods and proposing new algorithms and techniques. This research was mainly focused on spectroscopic data cubes, as they strongly require computational methods to reduce, visualize and infer astrophysical quantities from them, and because most of the techniques are directly applicable either to images or to spectra.

The ACALib philosophy is to use a persistent workspace abstraction where spectroscopic data cubes can be loaded from files, created from other cubes or artificially generated from astrophysical parameters. Then, computational methods can be applied to them, resulting in new data cube instances or new data tables in the workspace. The idea is to provide not only API bindings for the workspace and the cubes, but also webservices to use the library in cloud-based frameworks and in the Virtual Observatory.

In a nutshell, ACALib is integrating and testing several cube manipulation routines, stacking procedures, structure detection algorithms, spectral line association techniques and a synthetic data cube generation module. The library is developed in python, strongly rooted in astropy modules and using efficient numerical libraries such as numpy and scipy, and machine learning libraries like scikitlearn and astroML.

In the near future, we plan to propose ACALib as an astropy affiliated package, and to create a CASA add-on to ease the usage of our methods. Also, we are exploring bleeding-edge computational methods to include to ACALib, such as deep learning networks, and developing new prototypes for other types of astronomical data, such as light curves in the time-domain.

1. Overview

ACALib is a software package that implements several algorithms and tools for analyzing spectroscopic data cubes, images and spectra. The library consist in a coherent framework for developing novel webservices for processing data on-line in the Chilean Virtual Observatory (Solar et al. (2014)), but it also offers a generic API for developing stand-alone applications. The algorithms automatically connects to VO services (Araya et al. (2015a)), and the library is compatible with the SAMP protocol for interacting with other applications like Topcat or Aladin (see Figure 1:left).

The library is divided into 5 modules. The *core* module has the main classes that represent and manipulate astronomical data. The *VO* module provides the workspace abstraction and VO communication interfaces. The *synthetic* module generates synthetic data (spectral lines, flux distributions, meta-data, etc). The *process* module contains the algorithms developed so far by ChiVO. At last, the *graphic* module will have the widgets and tools for 3D visualization of spectroscopic cubes, clumps, surfaces, spectra, etc.

The rest of the paper describes these modules which were (or are been) developed reusing as much as possible from existing libraries. Indeed, the library is strongly based on Astropy (Robitaille et al. (2013)) and uses vectorized computations through numpy and advanced algorithms from scipy and scikit-learn.

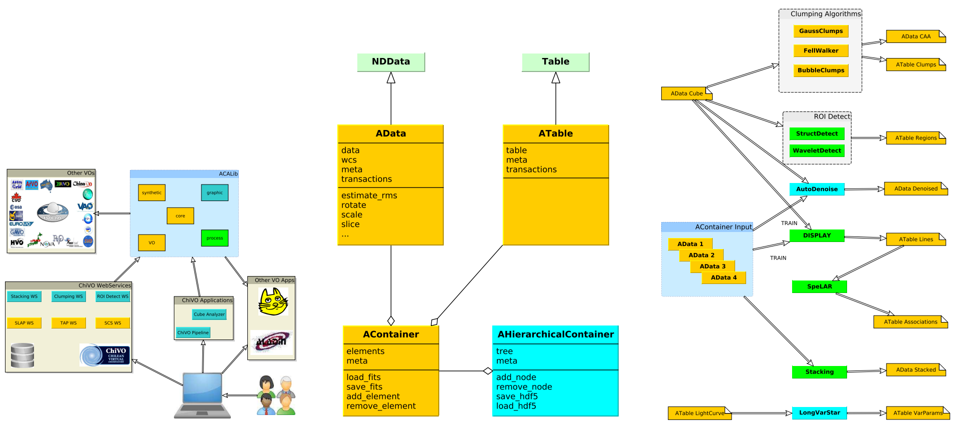

Figure 1. Module views of ACALib. Through the paper, the boxes in yellow are those components that are already implemented and integrated, the green ones are the ones that are already prototyped, but not integrated, and in blue/cyan are those that are currently under development. *Left:* Interactions of ACALib with other software. *Center:* Core module abstract data types. *Right:* Algorithms of the process module and data structures.

2. Modules

2.1. Core Module

This module contains the abstract data types that are used by all the other modules. `AData` is an extension of astropy NDData class, that uses vectorized masked arrays for storing data, allowing fast computations with missing values support. It inherits metadata and WCS support, and adds several methods like rotate, scale, slice, stack, statistics, search, etc. It also implements transactions in order to maintain consistency of WCS and metadata, either using a deferred or online updates. **ATable** is an extension of astropy Table, that provides a simpler interface to fill it and provide some basic statistics (see Figure 1:center).

AContainer is an aggregator class composed by a list of AData and ATable objects. It works as a namespace for astronomical data, and allows to load and save FITS files (also a data container). We plan to build a **AHierarchicalContainer**, which will be a tree of AContainer objects for supporting the HDF5 format

2.2. Process Module

The Algorithms implemented so far are:

- **Clumping**: detect clumps in an AData. Similar to CUPID package (Berry et al. (2007)).
  - GaussClumps: Mixture of Gaussians fitting (Stutzki & Guesten (1990))
  - FellWalker: Agregation of hill-climbing paths (Berry (2015))
  - BubbleClumps: Clustering of small Gaussians (unpublished)
- **ROI Detection**: index multi-resolution regions of interest in an AData.
  - StructDetect: Morphological processing (Mendoza et al. (2015))
  - WaveletDetect: Multi-scale detection in Wavelets space (Gregorio et al. (2014))
- DISPLAY: Learn dictionaries for line detection (unpublished) a
- SpeLAR: Compute association rules for spectral lines (unpublished)
- Stacking: Automatic stacking of images (unpublished)

In Figure 1:right a summary of the input/output data types of the algorithms can be found.

2.3. Other Implemented Modules

Due to space constraints we do not include a detailed description or diagrams of the other modules. The **VO** module implements a workspace abstraction, which is a class that host elements of the core (AContainer, AData and ATable), send them through SAMP and obtain/export data from/to the VO. Also, it allows loading and saving the whole workspace to disk.

The **synthetic** module is an integrated version of ASYDO (refer to Araya et al. (2015b)), which can generate data for testing, training and validation.

3. Future Work

Currently, we are working in the **graphic** module. This is a key module to validate the results of each algorithm, specifically by offering volumetric visualization of spectroscopic data cubes.

We also are exploring the performance details of the library for implementing the required services in our Data Centre. Specifically, a Cython integration for less memory consumption than NumPy and an MPI integration for speeding up automatic pipelines are undergoing.

In terms of algorithms, we are still integrating the prototyped algorithms (green), but in parallel we are investigation how to denoise of images using deep autoencoders and implementing machine learning algorithms for long-term variable stars detection.

Acknowledgments. This work was partially financed by CONICYT through the FONDEF D11I1060 project and the ICHAA 79130008 project.

References

Araya, M., Solar, M., & Antognini, J. 2015a, New Astronomy, 39, 46

Araya, M., Solar, M., Mardones, D., & Hochfärber, T. 2015b, in Astronomical Society of the Pacific Conference Series, vol. 495, 57

Berry, D. 2015, Astronomy and Computing, 10, 22

Berry, D. S., Reinhold, K., Jenness, T., & Economou, F. 2007, in Astronomical Data Analysis Software and Systems XVI, edited by R. A. Shaw, F. Hill, & D. J. Bell, vol. 376, 425

Gregorio, R., Solar, M., Mardones, D., Pichara, K., Parada, V., & Contreras, R. 2014, in Proc. SPIE, vol. 9152

Mendoza, M., Candia, G., Gregorio, R., Araya, M., & Solar, M. 2015, Astronomy and Computing, to appear

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., Lian Lim, P., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., & Streicher, O. 2013, A&A, 558, A33

Solar, M., Farina, W., Mardones, D., Antognini, J., Pichara, K., Nagar, N., Parada, V., Ibsen, J., Nyman, L., & Marroquin, J. 2014, in Proc. SPIE, vol. 9150

Stutzki, J., & Guesten, R. 1990, ApJ, 356, 513