An on-site data reduction pipeline for the Goodman Spectrograph

Simón Torres-Robledo,[1] and César Briceño[1,2]

[1]*SOAR Telescope, La Serena, Región de Coquimbo, Chile;* `storres@ctio.noao.edu`

[2]*Cerro Tololo Interamerican Observatory, Casilla 603, La Serena, Chile*

Abstract. The Goodman Spectroscopic Pipeline (GSP) is reaching some maturity and behaving in a stable manner. Though its development continues, we have started a parallel effort to develop a real-time version of the GSP, which aims at obtaining fully reduced data within seconds after the spectrum has been obtained at the telescope. Most of the required structure, algorithms and processes already exist with the offline version of the GSP. The real-time or online 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.

1. Introduction

The *SOAR Telescope* is a 4-meter telescope located on Cerro Pachón, northern Chile. The *Goodman High Throughput Spectrograph* (Goodman HTS) is an Imaging Spectrograph developed by the University of North Carolina, see Clemens et al. (2004). Is the most requested/used instrument currently at the telescope with about 65% of the time available on the sky. SOAR operations are run in classical mode, in which all approved proposals are scheduled on specific dates. Observers can go up to the summit, but most often they observe from a remote location, establishing an Internet connection through a VPN tunnel, and accessing the instrument software via a VNC connection to the appropriate computer desktop. 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. Following this path, a project has been set up in collaboration with NOAO and Las Cumbres Observatory (LCO), to automate 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) forms part of this project 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 on whether to obtain more spectra of a given object that may be on the rise, or fading. At the moment of writing GSP exists only as an offline version, working as a one-line command that can be run once the observing night has finished. The user uses a VPN tunnel plus VNC to connect to a dedicated data reduction machine on which we have the latest version of the GSP. Though VNC connections are relatively reliable and simple to setup they come with certain flaws, such as large bandwidth requirement, zero user management capabilities and with it, security flaws, non-responsive layout and usually requires the use of large screens. In this paper we describe the requirements for an implementation using web technology with secure authentication and responsive layout, using modern, well proven technology.

2. Status of development

Our offline data reduction pipeline 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)). Processing a typical full night takes between three and five minutes.

2.1. Wavelength Calibration

After experimenting obtaining wavelength calibrations with several methods, including through interactive manual input, we came to the conclusion that the best solution for our application was to use a catalog of templates, a collection of comparison lamp spectra taken with our own hollow cathode lamps. The Goodman spectrograph 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 coverages. 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. Still, if the user generates the templates, then the GSP will work with that specific setup. 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 Goodman spectrograph 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. Another technique we attempted is to use the grating equation in order to obtain a theoretical mathematical model, then using line list information, such as those from NIST, to build a template which then will be used to cross correlate with the comparison lamp obtained along with the science data that we want to calibrate. The problems with this method is that line list do not have accurate line intensity information and a common scenario found when trying to find out why it did not work is that the lines picked do not exist in the observed comparison lamp. This does not only have to do with the line intensity information but also the lamp itself is not perfect and also they evolve plus there is more than just a lamp involved in the process of obtaining a comparison lamp spectrum, see Sarmiento et al. (2018) for a list of other aspects that affect the resulting lamp spectrum.

Performance-wise the results are good, though again the precision will depend on many variables such as lamp quality, combination of grating and spectroscopic mode; this will affect the number of lines present in the lamp spectrum and ultimately the slit size. Overall, we mainly measure the solution RMS obtaining similar values to those obtained using IRAF but automatically and in a couple of seconds or less. For instance using the 930 l/mm grating with the mode M2 the RMS error obtained was 0.281.

2.2. Room for Improvement

There are still many areas where to improve, for instance, addition of a flux calibration module that should be used to create an exposure time calculator, removing DCR (Pych (2004)) which is a great tool for removing cosmic rays written in C but even though it was seamlessly integrated into the pipeline the installation adds more complication and cannot be installed using PIP. Also our goal was to use as much Astropy code as possible and this was one exception due to how well it behaved. Astroscrappy is an implementation of LACosmic (van Dokkum (2001)) which is wrapped by `ccdproc.cosmicray_lacosmic`. At the moment we have both of them running and we realized that is just a matter of fine tunning the parameteres in order to obtain the same results. Is not an easy task though.

3. Design Constraints

For the live data reduction pipeline we want to get rid of the VNC system and move to a web-based service, this adds a whole new level of complexity but the results are worth the effort, the offline version of the pipeline exists as a single Python package but for the live version we will need several other components.

3.1. Data Reduction Package

This is based on the offline Python package. The modifications required are intended to enable asynchronous data processing as well as controls, this could be achieved by adding a private REST API that would allow to modify 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 should 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 itself most of the routines are already implemented in the offline version but there are a couple of things that are still unimplemented such as optimal extraction [Marsh (1989), Horne (1986)], flux calibration, deblending of multiple sources, low signal-to-noise extraction and so many more.

As is explained in Torres-Robledo et al. (2017), 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 that we had to develop because they were not implemented in Astropy will be replaced when they become available or even better we will add them 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 the application has to be stable and reliable as well. Lucky for us there are plenty of tools that allow us to simplify these tasks, probable most of them. Since we don't have the resources to hire an entire team of developers already experts on these very specific tools, we will have to do it with the resources we have. The scope of the services provided by the public API should be end user oriented only, such as, secure authentication as well as channeling communications with the private API.

3.3. Web Front End

A highly responsive website will be favored over a local GUI for one simple reason: Most of the Goodman HTS users are working remotely and local users can still benefit from it. Though is more difficult to implement there are several benefits that comes with it, for instance: adaptive layout, user experience less dependant on connection quality, less bandwidth usage and of course taking advantage of the interactive experience that web technology allows.

We have not decided what are the tools (stack) that we are going to use, but the criteria for selecting them are: Have to be well documented and easily testable, by *easily* we mean that there have to be good tools 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

Pych, W. 2004, PASP, 116, 148. astro-ph/0311290

Sarmiento, L. F., Reiners, A., Huke, P., Bauer, F. F., Guenter, E. W., Seemann, U., & Wolter, U. 2018, A&A, 618, A118. 1806.07361

Torres-Robledo, S., Briceno, C., Quint, B., & Sanmartim, D. 2017, in ADASS XXVIII, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

van Dokkum, P. G. 2001, PASP, 113, 1420. astro-ph/0108003