



# A nearreal time Data Reduction Pipeline for the Goodman High Throughput Spectrograph



Simón Torres-Robledo<sup>1</sup>, César Briceño<sup>1,2</sup>.

<sup>1</sup>SOAR Telescope, La Serena, Región de Coquimbo, Chile.

<sup>2</sup>Cerro Tololo Interamerican Observatory, Casilla 603, La Serena, Chile.

## Abstract

The *Goodman Spectroscopic Pipeline* (`goodman-pipeline` or 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 to 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.

## Highlighted Features

The latest version of the Goodman Spectroscopic Pipeline (GSP) is the result of extensive development over two years. Our initial goals were:

- Easy to use
- Well documented
- Written according to current standards
- Capable of producing science-quality results

In addition to meeting these goals, the GSP has the following attributes:

- Wavelength solution: High precision, repeatable.
- Modular: Part of the code can be imported for reuse.
- Integrated test: Automatically tested everyday.
- pip-installable: Since version v1.1.2
- Portable: The pipeline works on Linux and Mac.

## Current Performance

Speed has not been a top requirement for the GSP. This pipeline is specific for an instrument for which the data production rate is modest and easily manageable with existing resources. The focus of our effort has been more on obtaining good wavelength solutions. We found that the most practical way for obtaining accurate and repeatable wavelength solutions is to have a template library, i.e. wavelength calibrated lamps for all possible instrument configurations. The *Goodman HTS* has 20 fixed instrument configurations and six possible comparison lamps choices which gives a total of 120 lamps required for the template library to be complete.

## Trying it out

Because we don't have staffing resources to provide technical support for installation, we have set up a server located in the telescope computer room that users can access via a VNC session. However installation is easy:

```
pip install goodman-pipeline
```

This will install everything except DCR, the cosmic ray rejection routine, which you can install following the instructions in the documentation. The full documentation can be found at <https://goodman.readthedocs.io> or by scanning the QR code below.

## GSP Live Version: Conceptual Design and workflow

The live version of the pipeline has not been built yet, however we have experimented with parts that we are planning to use. From the technical point of view we don't see any particular challenge that we did not face already. All the processing will be done using the existing pipeline with the exception of the Data Quality assessment part. We have not decided yet whether it will be built as a submodule of the current project or as an independent project. It will work by actions

triggered by file-system events. For instance, a new file is added to the folder being monitored, the Hub (See Figure 1) is notified and the file is copied over to a local directory. Finally, the appropriate actions are triggered in order to ensure that the file is processed accordingly, this includes building master flats or master bias, show a preview of the current processed file or waiting for more images for combination.

## New Dependencies

**Pyinotify:** Is a filesystem events monitor, it is better than others because it notifies when the file has been closed after writing, i.e. file is complete.

**Zmq:** Is a high performance asynchronous messaging library, it allows to communicate processes using many languages and protocols.



ReadTheDocs



GitHub

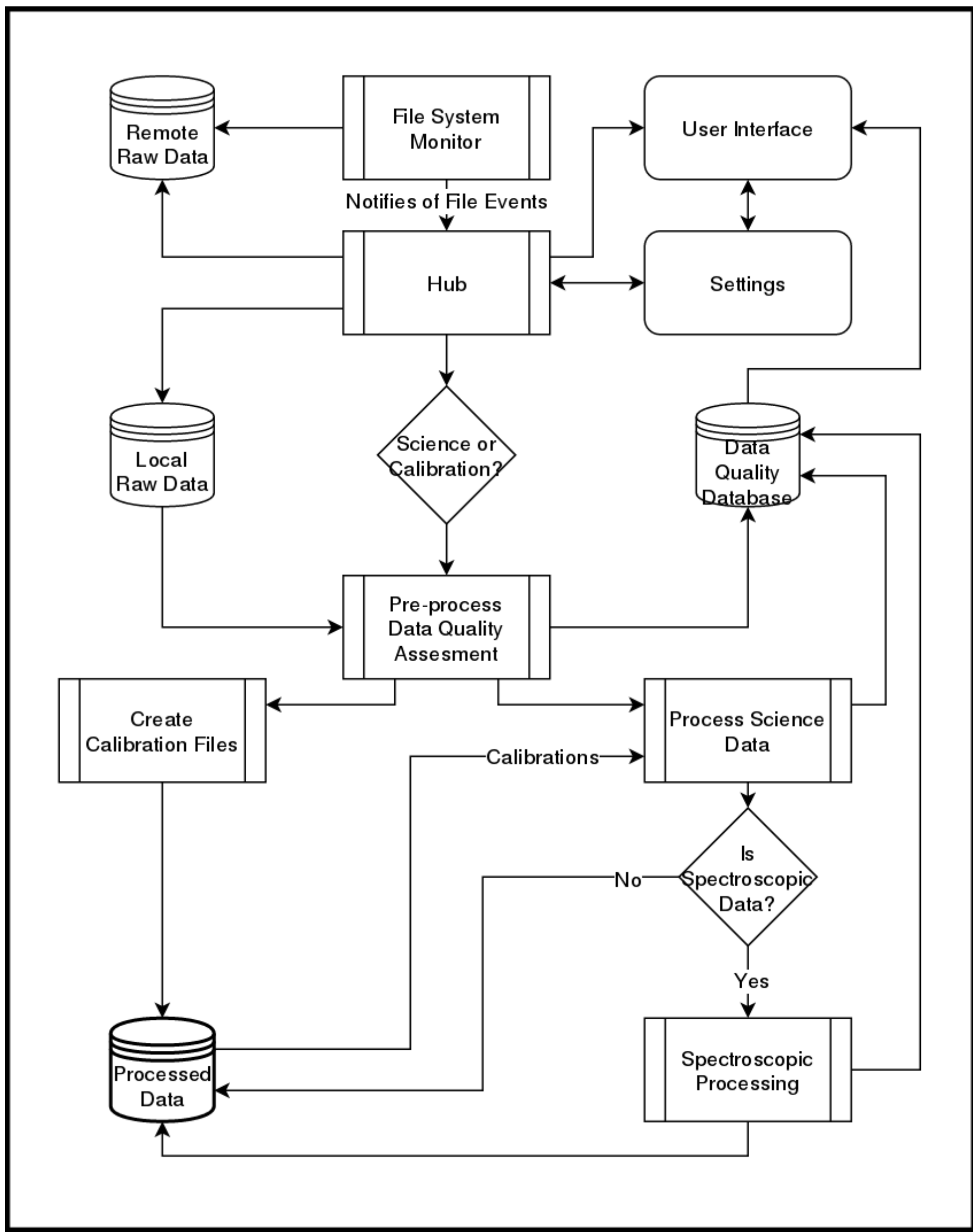
## Challenges

**User Interface:** There are quite a few options for doing the UI; web based using Django, or any of the UI development tools, such as Qt. Django and Qt have good Python integration, but since nothing has been done yet, an even better alternative may come up.

**Metrics and Data Quality Assessment:** Currently we do not have much on measuring the incoming data quality and for live data reduction this is extremely important.

**Process Control:** We have developed a working pipeline for data reduction at the end of the night, the requirements for an on-site reduction might differ in ways we can't really preview. Also different projects might have different approaches to data reduction.

**Faint Targets:** We have intentionally left faint target extractions under-developed because we assumed they would show up in minor amounts, but at the telescope the observer might want to see a preview of a faint object's spectrum.



**Figure 1 .** A proposed workflow schematic for the real time pipeline. It could work on a single computer or two machines communicated via a TCP/IP link.

## Conclusion

We have developed a conceptual design for a live version of the Goodman Spectroscopic Pipeline. It will use most of the routines already developed for GSP for data processing, but it will also require other tools to be developed, such as an user interface and some robust data quality assessment routines.

We continue committed to developing open source software using as much Astropy-affiliated packages as possible. Though we have been discussing the creation of the live version of the GSP for quite some time, this is our first approach to putting everything together. There are many iterations to go through, but ultimately it will be the final users that will define, in great measure, what the end product will *look* like.

## Contact

Simón Torres Robledo  
SOAR Telescope  
Email: storres@ctio.noao.edu  
Website: <https://github.com/simontorres>  
Phone: +56 51 2205348

## Acknowledgments

The authors would like to acknowledge the important contributions from Bruno Quint, David Sanmartín and Tina Armond.

This research made use of Astropy, a community-developed core Python package for Astronomy (Astropy Collaboration, 2013 and 2018).

This work has been developed at the Southern Astrophysical Research (SOAR) telescope, which is a joint project of the Ministério da Ciência, Tecnologia, Inovações e Comunicações do Brasil (MCTIC/LNA), the U.S. National Optical Astronomy Observatory (NOAO), the University of North Carolina at Chapel Hill (UNC), and Michigan State University (MSU).

## References

1. Clemens, J. C., Crain, J. A., & Anderson, R. 2004, vol 5492 of Proceedings of SPIE, 331.
2. Astropy Collaboration et. al., 2013, A&A, 558, A33.
3. Astropy Collaboration et. al., 2018, AJ, 156, 123.
4. Marsh, T. R., 1989, PASP, 101:1032-1037.
5. Pych, W., 2004, PASP, 116, 148.
6. Van Dokkum, P. G., 2001, PASP, 113, 1420.