Development of Data Reduction Pipelines for GTC Instruments at the UCM

S. Pascual, J. Gallego, N. Cardiel, M. C. Eliche-Moral
Departamento de Astrofísica y CC. Atmósfera, Facultad de CC. Físicas, Universidad Complutense de Madrid, 28040 Madrid, Spain
http://guaix.fis.ucm.es sergiopr@fis.ucm.es

The UCM Instrumentation Group (GUAIX) currently is developing **Data Reduction Pipelines** (DRP) for four instruments of the GTC: **EMIR**, **FRIDA**, **MEGARA** and **MIRADAS**.

The purpose of the **DRP**s is to provide astronomers **scientific quality data**, removing instrumental biases, calibrating the images in **physical units** and providing a **estimation of the associated uncertainties.**

The DRPs will be provided as **stand-alone packages**, independent of the GTC Control System. However, we are working together with the GCS developers to **integrate the DRPs into the GTC Control System**, providing online data processing capabilities for those observing modes that are required to complete successfully the observations.

Development Overview

We are developing all our processing tools using Python. The development is compatible with **Python 2.7.** The code is stored in several public Mercurial repositories under **https://guaix.fis.ucm.es/hg**

Different Python packages provide functionalities we require: **Numpy**, **Scipy**, **Pyfits**, **Pywcs**, **Matplotlib**, **Sphynx**, **Nose** and **Cython**.

All the developed packages are be released under GPLv3. They **will be free for everyone to install, share and modify.**

Parts of the code (extensive numeric operations within arrays) are **implemented in C++** for efficiency, using either **Cython** or manually using the Python/Numpy API .

Architecture and Packages

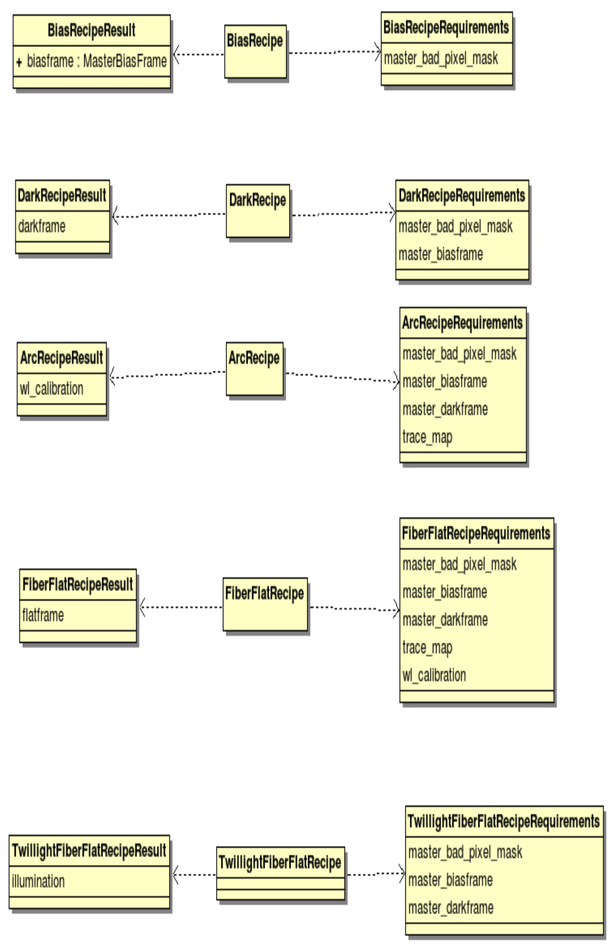
The logic of the reduction is stored in **Recipes.** Each instrument pipeline package contains the reduction recipes for all **Observing Modes** of the instrument.

Each Recipe provides a list of its **Products**. Products can be numeric or images. Each product belongs to a given class of data products.

Each Recipe contains a list of its **Requirements**. Requirements can be numeric arguments or **Data Products** of other recipes.

The Recipe is not capable of searching for its requirements. They have to be provided by a higher level software component.

Recipes implement a **plugin interface** so that they can be loaded from the system given the name of the observing mode they process. We provide such a recipe loader, with a simple CLI in the **common services packages Numina.**

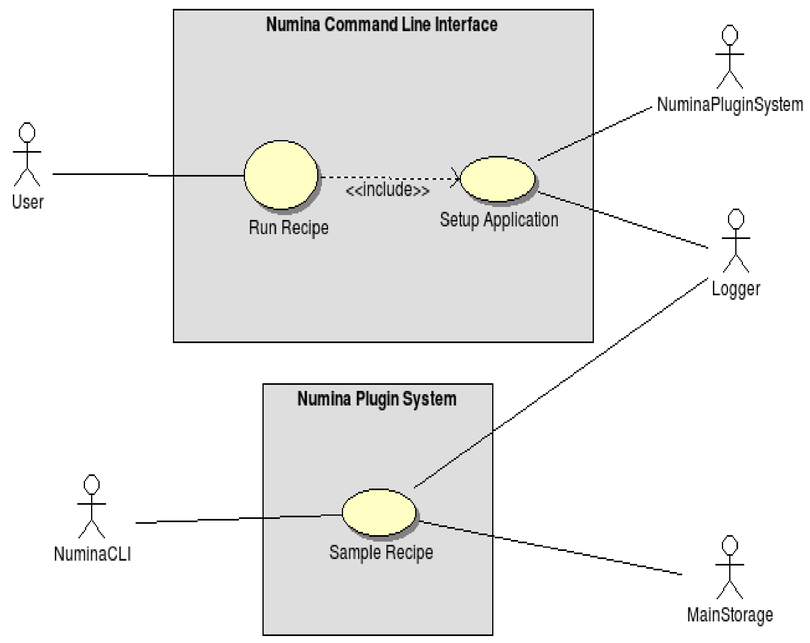


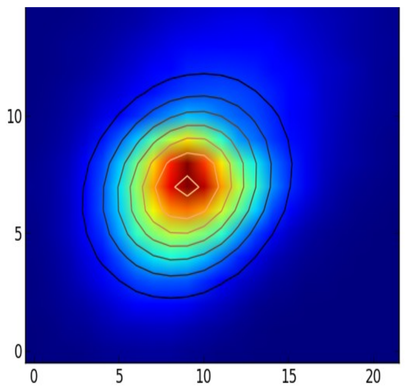
Numina: Common Services Package

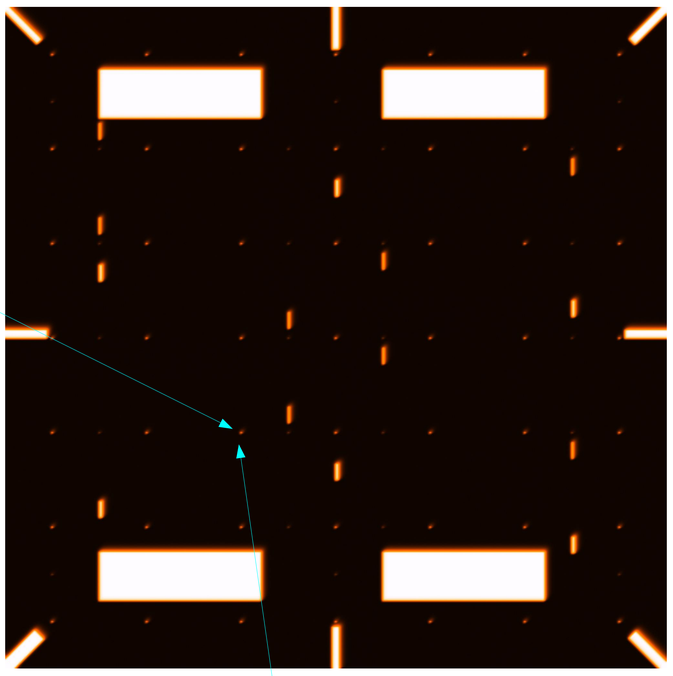
The common services package includes functions and classes useful for all the pipelines: base classes for Recipes, plugin API, etc.

With Numina, we provide a basic recipe execution program, so that the user is able to run recipes by hand. The command line tool is in charge of loading the recipe, configure it using configuration files and run it with available data in the disk. The components of the tool are shown bellow.

Based on the plugin architecture of the recipes, more complex recipe runners can be created **without modifying the Recipes.** Our group has experimented with database backends and automatic processing.







Status of the pipelines

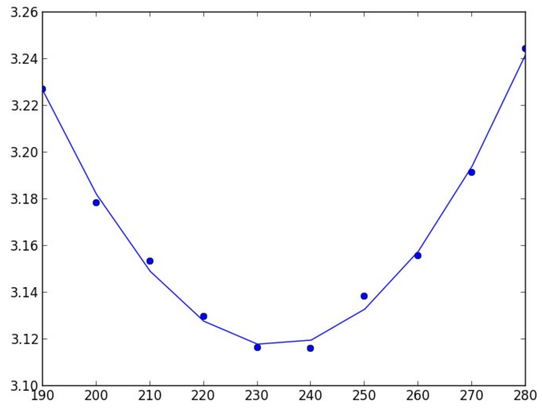
FRIDA DRP is designed and ready to be developed.

MEGARA DRP is in the final stages of design.

MIRADAS DRP is in the early stages of design.

EMIR DRP is in the final stages of development. We are providing tasks to be run during AIV, such as “best focus determination”, “linearity”, “gain”, “readout noise”, etc.
These tasks will also help to test the integration between the GCS and the DRP.

As an example we show part of the outputs of a task used to determine the best focus on a simulated pinhole mask.



Integration with the GTC Control system

The plugin architecture of the DRPs allow to build different Recipe loaders without changing the Recipes and the logic of the data reduction.

The GTC Control group, the EMIR Control group and the GUAIX group are working together in a solution to integrate the EMIR DRP into the GCS. The development shall be readily used by other instruments. This solution passes through the development of a component that, on one hand, interacts with the GCS as a native Data Factory branch and, on the other hand, loads and controls the Python Recipes.

This component will be written in Python and will communicate via CORBA with the rest of the GCS