PyEmir: Data Reduction Pipeline for EMIR the GTC Near-IR Multi-Object Spectrograph

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/projects/emir
sergiopr@astrax.fis.ucm.es

Gran Telescopio Canarias

The 10.4m Gran Telescopio Canarias (GTC) is located in one of the best observing sites in the Northern Hemisphere, Observatory Roque de los Muchachos (La Palma, the Canary islands, Spain). The telescope primary mirror is made of 36 segments and it has been designed to work in the optical-nIR range.

Python EMIR pipeline

- The EMIR Data Reduction Pipeline is being developed in **Python** by developers of the **Guaix group** at the UCM. Several reasons led us to change from our previous approach (based on a GTC C++ library) to a Python solution, namely:
  - The ability to create **interactive processing** sessions using the same tools used in offline processing (astronomers are very used to fine tune parameters in interactive sessions).
  - Python is being used in multiple astronomical projects.
  - Python is object-oriented and interfaces with C/C++. We can reuse part of our previous development.

Reduction recipes

- Recipes are the basic unit of data processing.
- Each instrument has a list of predefined **observing modes** (OM).
- Each OM has a Recipe prepared to process the data. It defines also the metadata needed for the reduction

Common Interface

- Recipes implement a **plugin interface** so that they can be loaded from the system given their name.
- An application (recipe runner) is in charge of loading the recipe, configure it and run it with the appropriated data.

EMIR Data Reduction Pipeline

- The Data Reduction Pipeline (DRP) will process data acquired with EMIR. It shall deliver reduced images, by means of robust software tools and algorithms optimized for handling near-IR data.
- The DRP is prepared to handle wide-field frames in imaging mode, as well as, multi-object observations (up to 45 slitlets).
- The frames obtained with EMIR shall be reduced into scientific images and delivered to astronomers.
- Some of the feature included are:
  - Check and quality control procedures
  - Error propagation is considered.
  - The reduction process is fully automatic

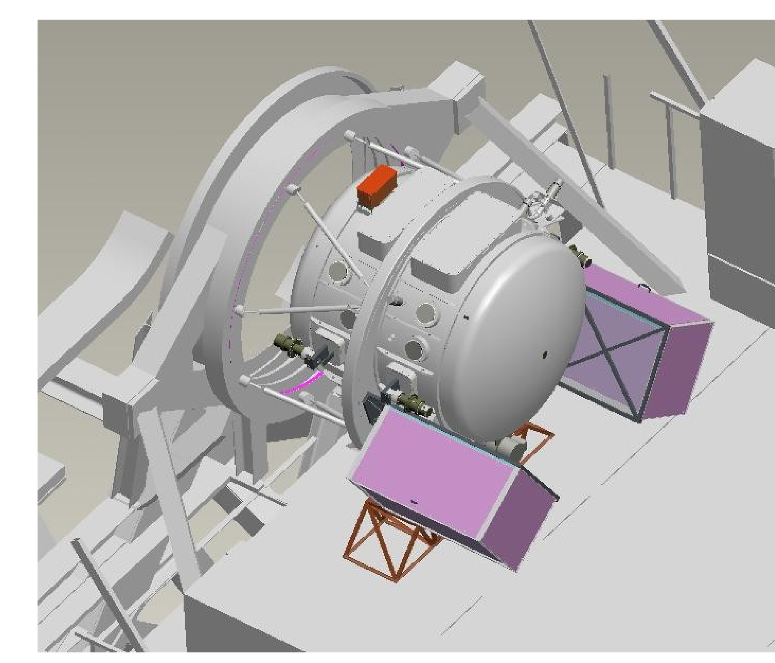
The PyEmir package

- The development is compatible with Python 2.5
- Several python packages provide different functionalities:
  - Scipy (multidimensional arrays)
  - Nose (extended unit testing)
  - Setuptools (installation)
  - PyFits (FITS files handling)
- The Python EMIR DRP (PyEmir) will be released under GPLv3. **It will be free for everyone to install, share and modify.**
- The code is documented using Sphynx, online and offline documentation of the pipeline will be provided in different formats (html, PDF).
- Parts of the code (extensive numeric operations within arrays) are **implemented in C++** using the Python/Numpy API for efficiency.
- Our plan is to provide a development environment for other GTC instruments. We divide our package in two modules.
  - **Module emir** contains Emir specific code and processing routines.
  - **Module numina** contains classes to develop compatible reduction recipes, the command line recipe runner, logging capabilities, generic image manipulation, etc.

```python
__version__ = "$Revision: 501 $"

# Classes are new style
__metaclass__ = type

class RecipeBase:
    '''Abstract Base class for Recipes.'''
#    __metaclass__ = abc.ABCMeta
    def __init__(self, optusage=None):
        if optusage is None:
            optusage = "usage: %prog [options] recipe [recipe-options]"
        self.cmdoptions = OptionParser(usage = optusage)
        self.cmdoptions.add_option('--docs', action="store_true", dest="docs",
                                   default=False, help="prints documentation")
        self.iniconfig = SafeConfigParser()
        self._repeat = 1

    def setup(self):
        '''Initialize structures only once before recipe execution.'''
        pass

    def cleanup(self):
        '''Cleanup structures after recipe execution.'''
        pass

    def run(self):
        '''Run the recipe, don't override.'''
        try:
            self._repeat -= 1
            result = self.process()
            return result
        except RecipeError:
            raise

#    @abc.abstractmethod
    def process(self):
        ''' Override this method with custom code.

        :rtype: RecipeResult
        '''
        raise NotImplementedError

    def complete(self):
        '''True once the recipe is completed.

        :rtype: bool
        '''
        return self._repeat <= 0
```

EMIR in a nutshell

The *Espectrógrafo Multi-Objeto Infrarrojo* (EMIR) is a state-of-the-art near-infrared wide-field camera and multi-object spectrograph which is designed to work at the Nasmyth focus of the 10.4m Gran Telescopio Canarias.



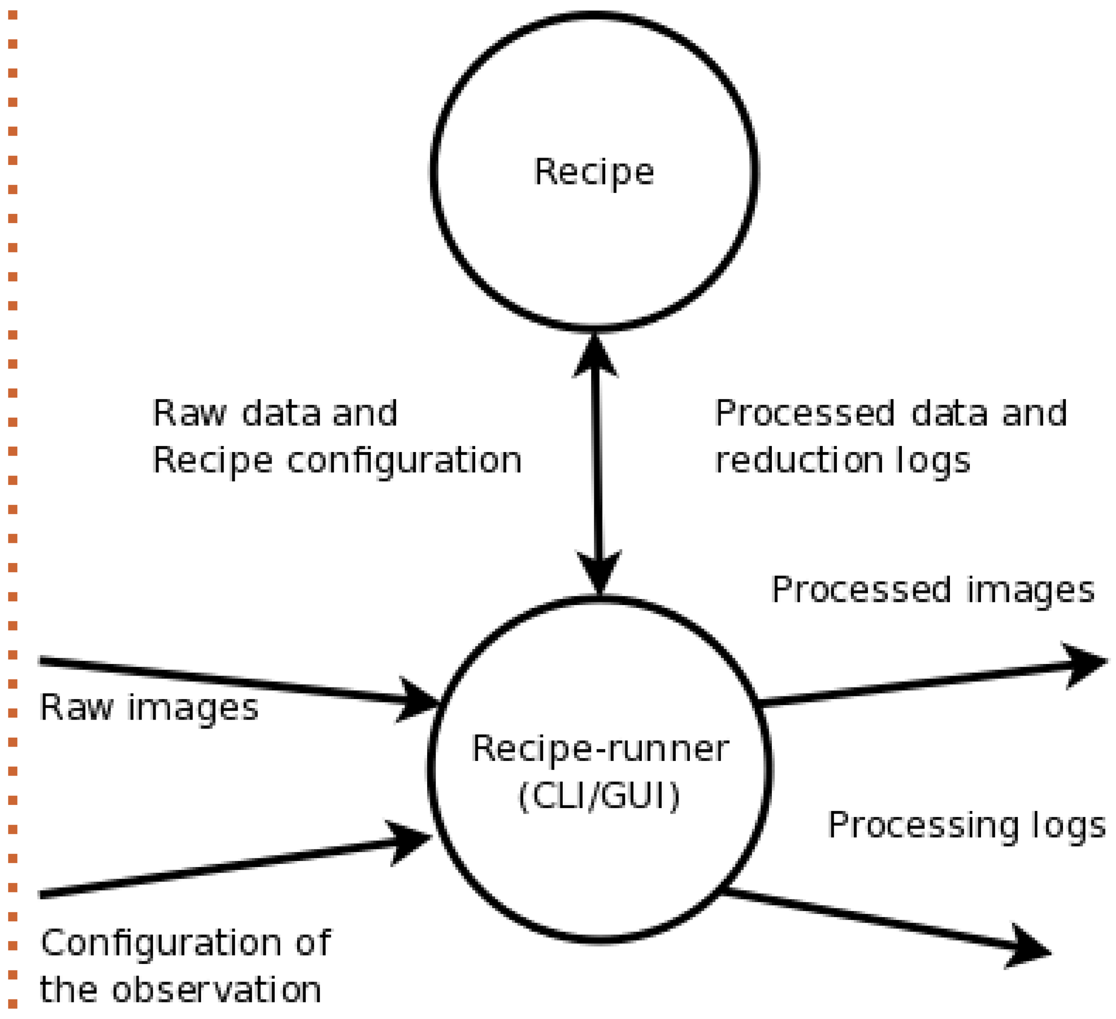
The Guaix group at the UCM

The **Extragalactic Astrophysics and Astronomical Instrumentation group of the Universidad Complutense de Madrid** is composed of more than 20 astronomers.

The research interests of the group include **stellar populations** in dwarf and elliptical galaxies, local **star forming galaxies** and **cosmological surveys** and the development of astronomical **software and instrumentation.**

For more information, please visit: **http://guaix.fis.ucm.es**

Data flow diagram of a user- driven reduction



The recipe runner (either command line or GUI) is in charge of handling the raw data and the files describing the observation. The recipe runner will select the suitable reduction recipe based on the **observing mode** of the observation.
The recipe will process the data and will return the processed data to the runner. The runner will save the results in the appropriated place.