SIPGI: an interactive pipeline for spectroscopic data reduction

A. Gargiulo, M. Fumana, S. Bisogni, P. Franzetti, L. Cassarà,
M. Scodeggio, B. Garilli, G. Vietri
INAF - IASF Milano

Outline

- What is SIPGI;
- the concept behind SIPGI;
- the three SIPGI pillars;
    - the graphical interface
    - the instrument model
    - the recipes organization
- the SIPGI performances;
- SpectraPy

What is SIPGI

SIPGI → **S**pectroscopic **I**nteractive **P**ipeline and **G**raphical **I**nterface

A complete spectroscopic reduction environment for optical and near-IR through-slit spectra with an

high level of flexibility *and* efficiency

SIPGI is descended from VIPGI (*Scodeggio et al. 2005*),
the VIMOS spectroscopic pipeline we developed and used for the official reduction of the major VIMOS spectroscopic extragalactic surveys.

(e.g. **VVDS, zCOSMOS, VUDS, VIPERS, VANDELS** - *Garilli et al. 2008, Le Fevre et al. 2013, Lilly et al. 2007, Le Fevre et al. 2015, Guzzo et al. 2014, Garilli et al. 2014, Scodeggio et al. 2018, McLure et al. 2018, Pentericci et al. 2018, Garilli et al. 2021*)

What SIPGI is

SIPGI inherits the main concepts of VIPGI and extends reduction recipes to the **near-infrared**.

↓

We customized SIPGI for the optical LBT/MODS and near-IR LBT/LUCI spectra → LBT spectroscopic reduction center @IASF-Mi

↓

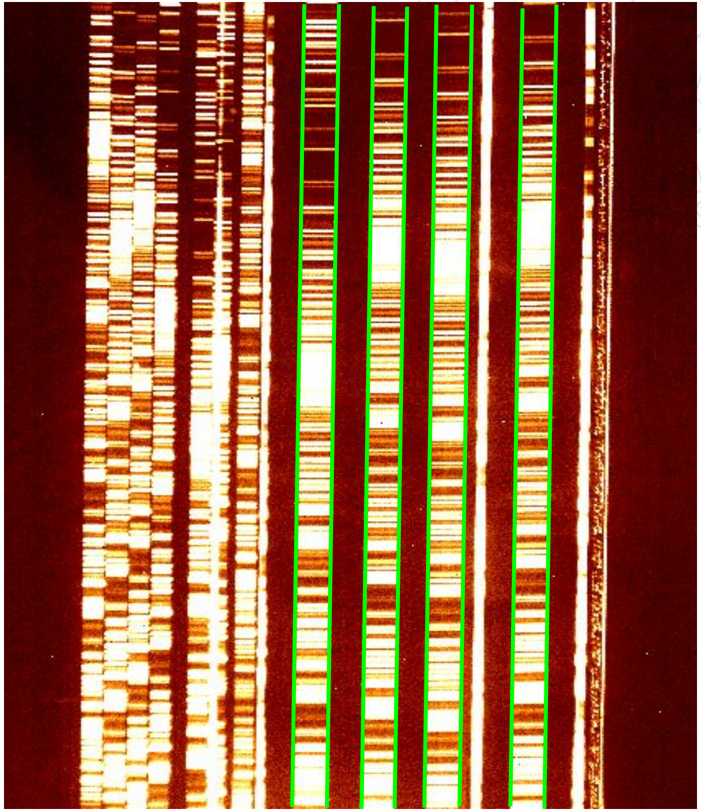
We released this LBT-customized SIPGI version

(Gargiulo et al. submitted)

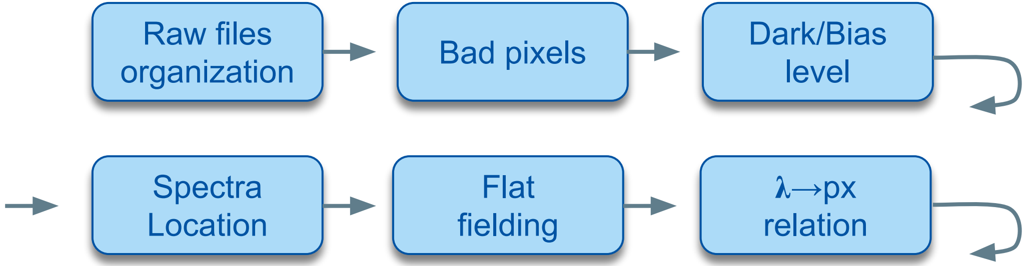
The main spectroscopic data-reduction steps

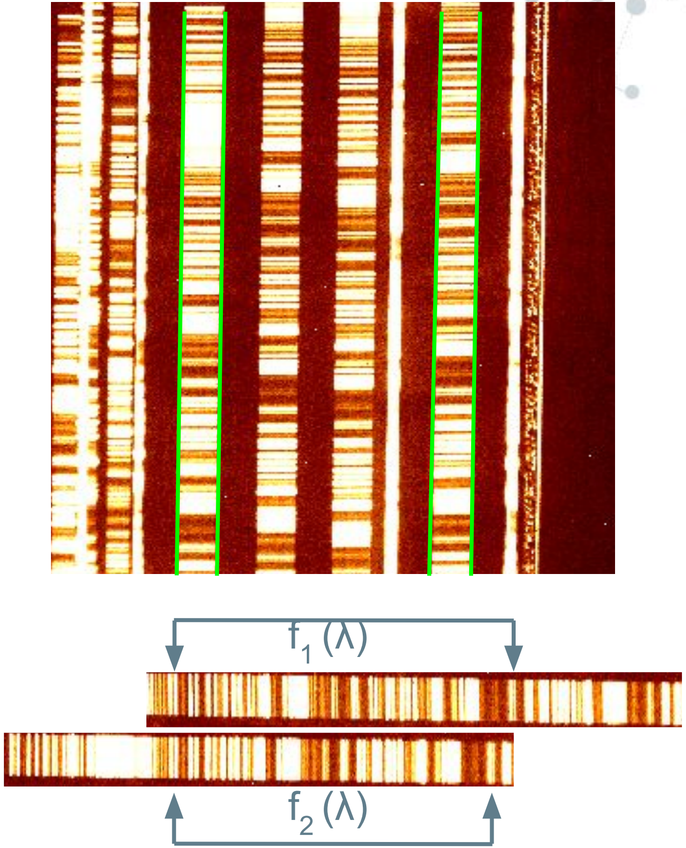
Raw files organization → Bad pixels → Dark/Bias level →

→ Spectra Location



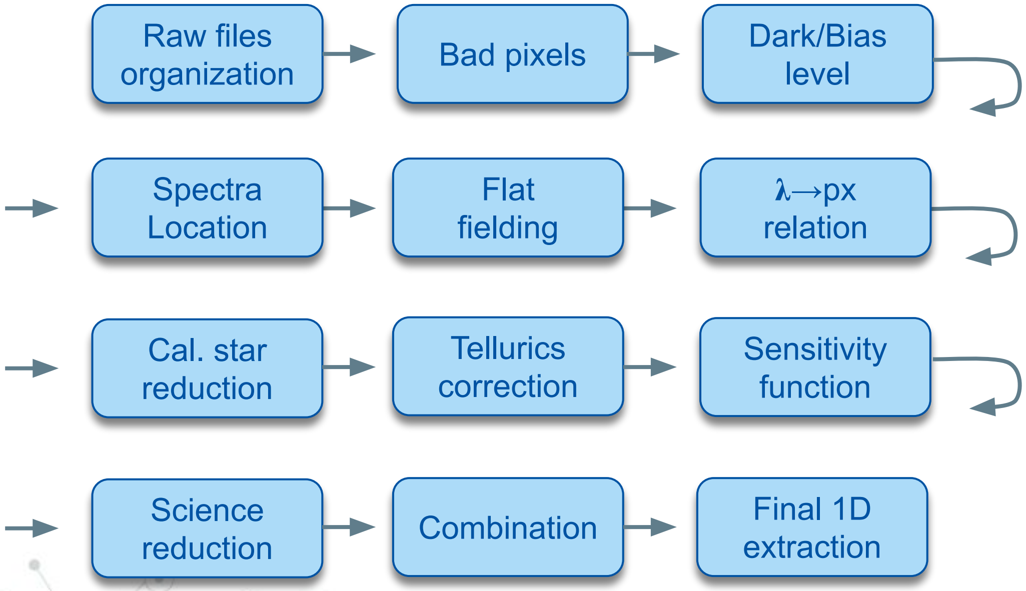
The main spectroscopic data-reduction steps



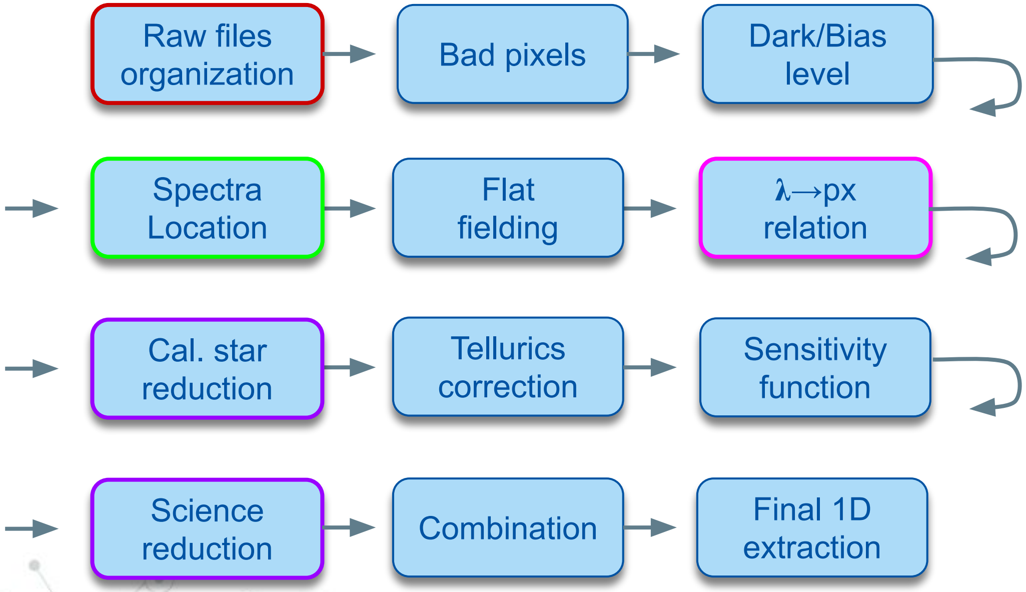


Both spectra location and λ→px relation must be estimated for each mask of the scientific project.

The main spectroscopic data-reduction steps



The concept behind SIPGI

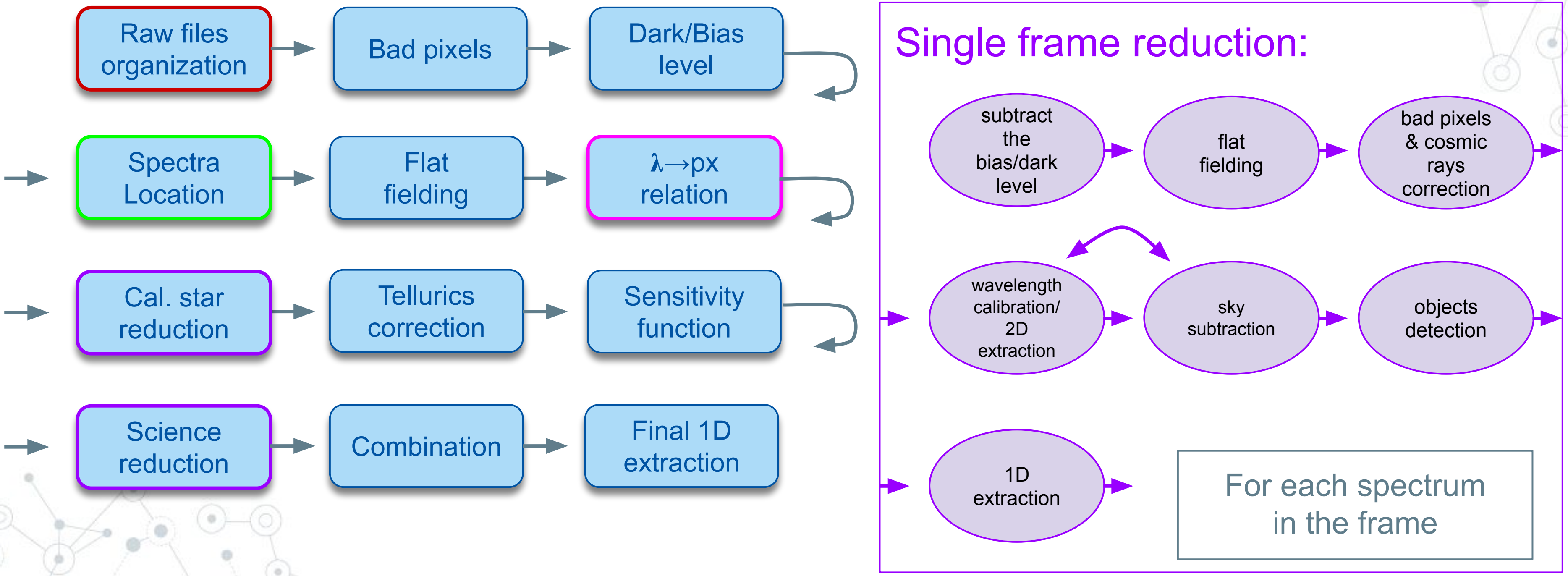


Raw files organization:
identify science/star/lamp/flat frames;

Spectra Location:
identify the location of *each* spectrum;

λ →px relation:
wavelength calibrate *each* slit;

The concept behind SIPGI



The three main SIPGI pillars

1. **The built-in data organizer and the graphical interface;** Raw files organization

2. **The instrument model;** Spectra Location | λ→px relation → MOST INNOVATIVE SIPGI ASPECT

3. **The recipe organization** Frame reduction

The three main SIPGI pillars

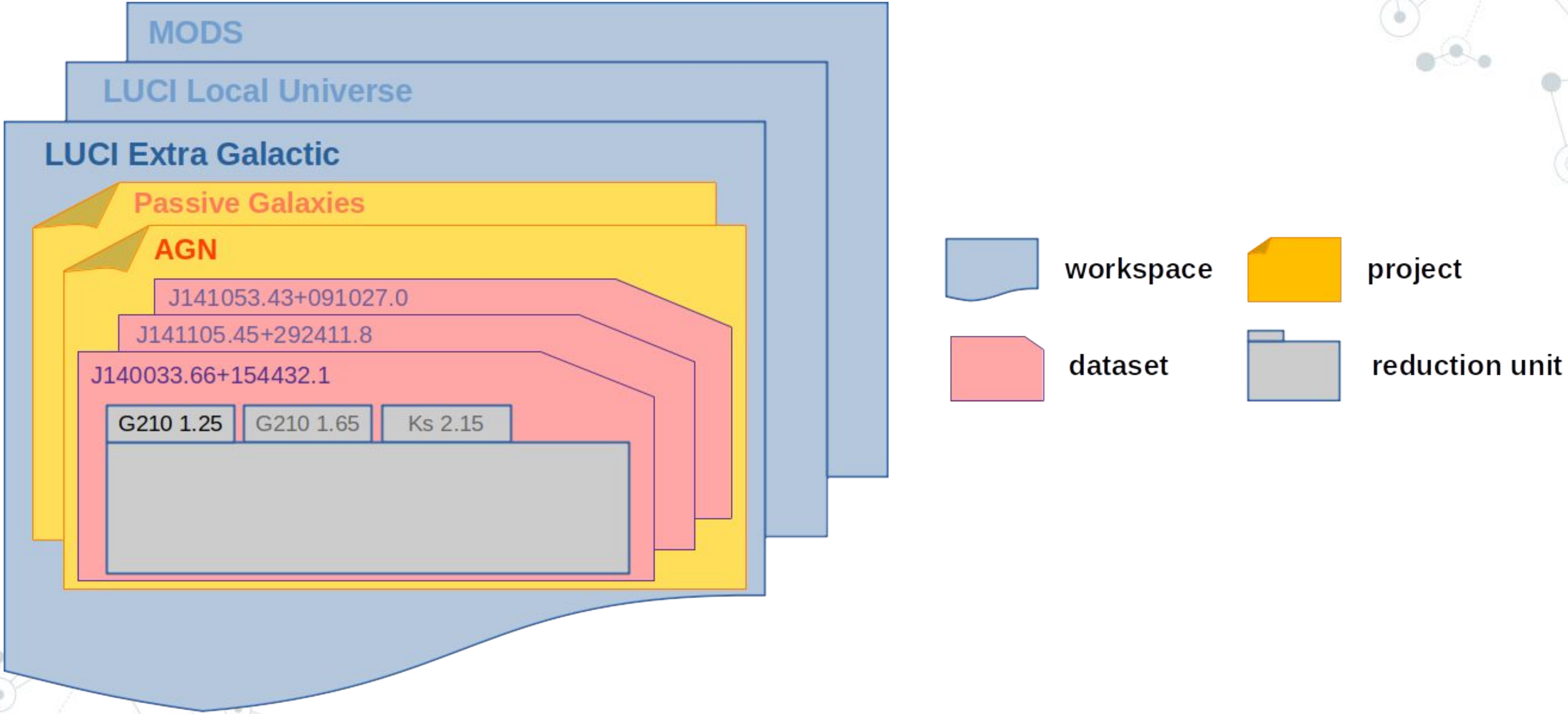
Raw files organization

1. The built-in data organizer and the graphical interface

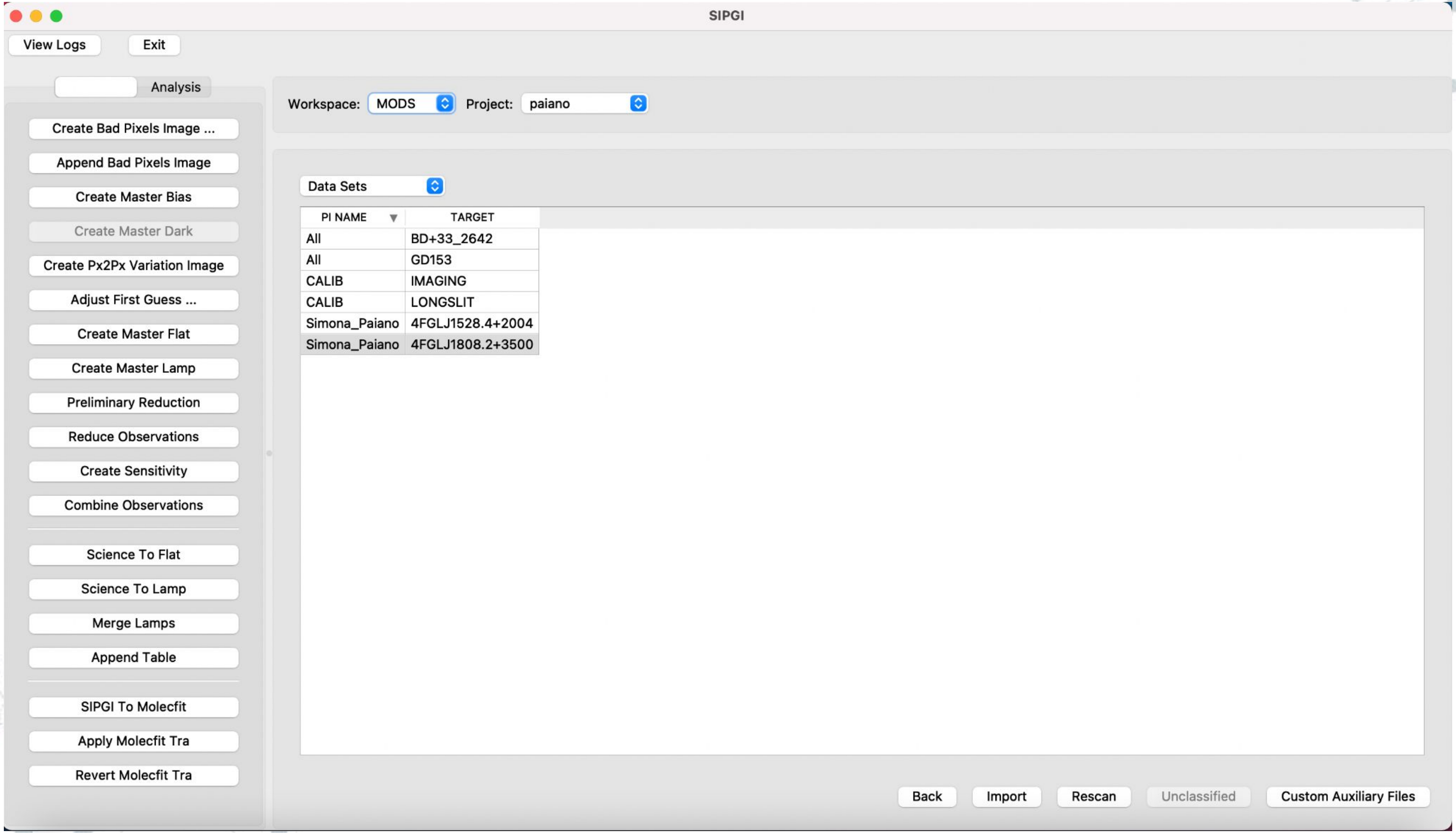
- Sipgi ingests raw data and, using FITS header keywords, categorizes them according to their data-type (bias/dark/flat/science/etc);

- the graphical interface allows to easily browse across the categorized data and to provide the right input to the right recipe.

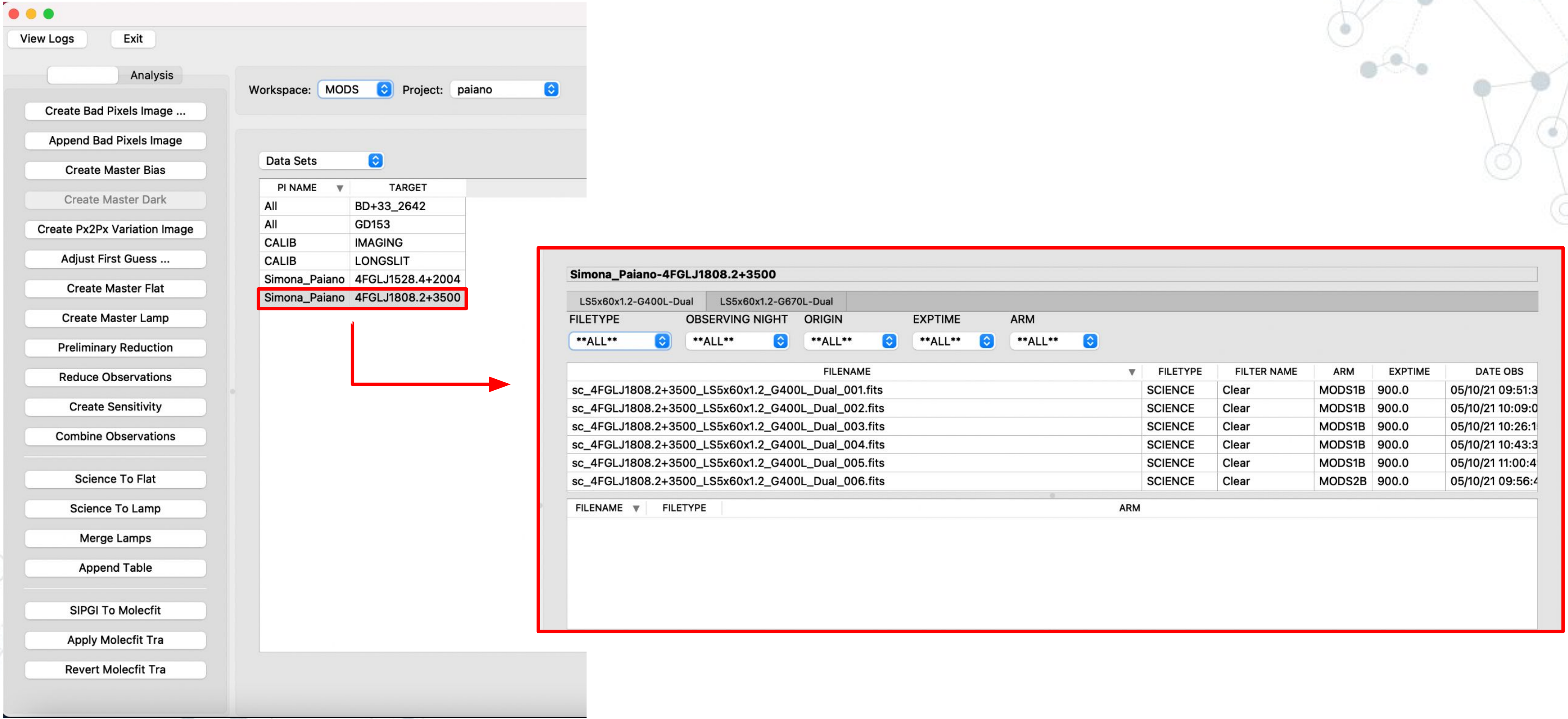
The three main SIPGI pillars - *the organizer*



The three main SIPGI pillars - *the graphical interface*



The three main SIPGI pillars - *the graphical interface*



The three main SIPGI pillars

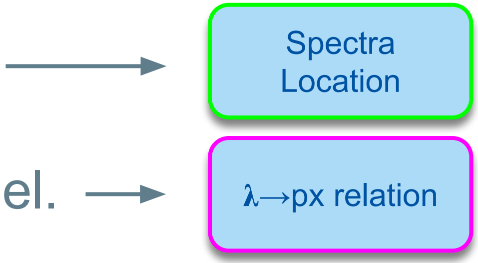
2. The instrument model

- an analytical description of the main calibration relations:

  - the **slit location** model;

  - the **spectra location** model;

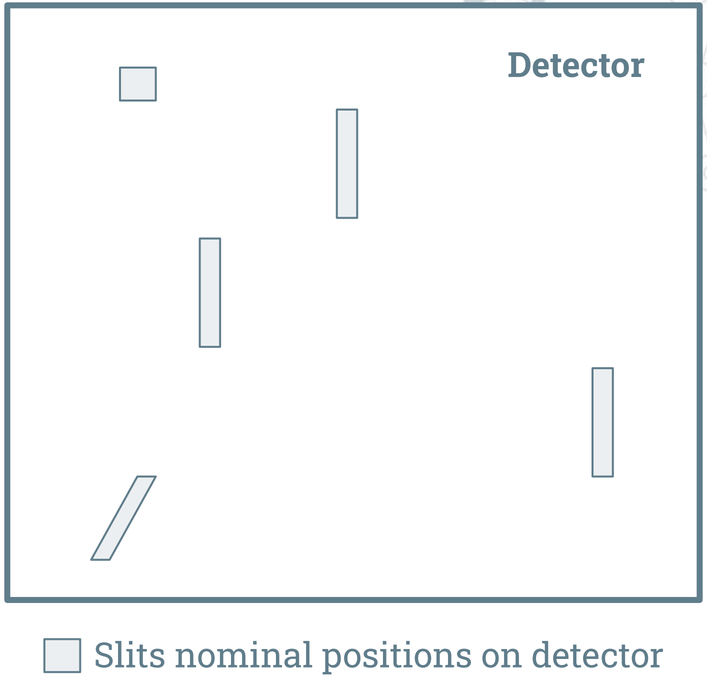
  - the **lambda calibration** model.



The three main SIPGI pillars - *the instrument model*

2.1 The slit location model

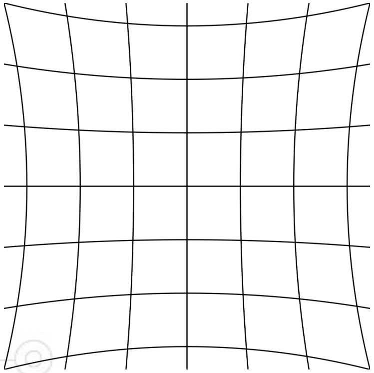
Given the instrument configuration (e.g. grism, mask, detector…), it is possible to predict the nominal positions of the slits on the detector.



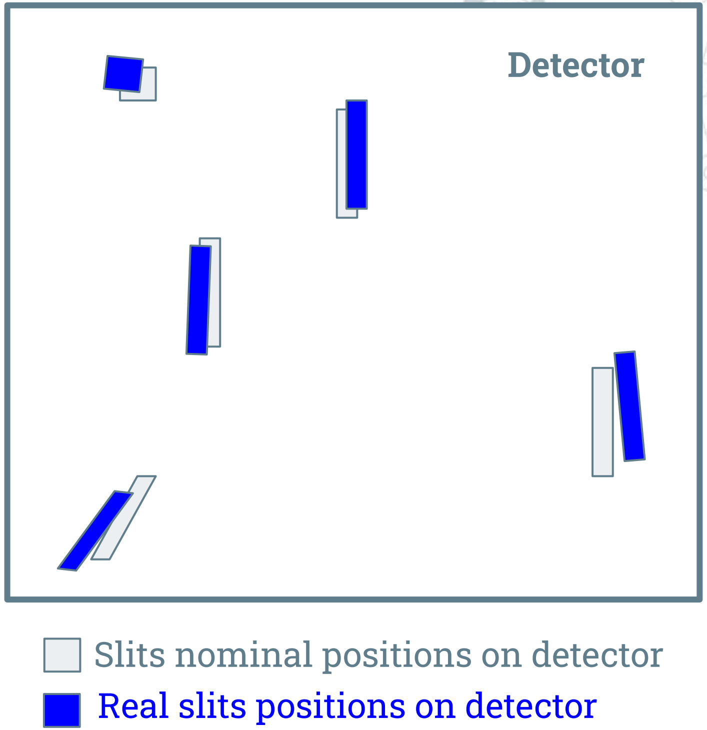
The three main SIPGI pillars - *the instrument model*

2.1 The slit location model

Instrument **optical distortions** affect these positions.



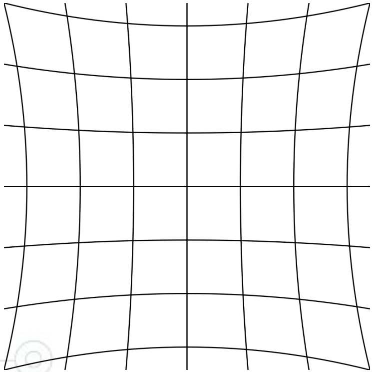
Instrument optical distortions representation



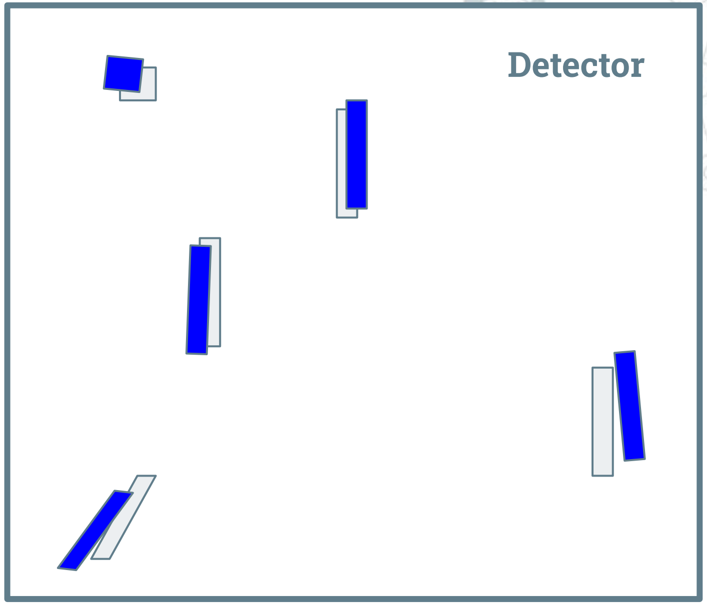
The three main SIPGI pillars - *the instrument model*

2.1 The slit location model

The **slit location model** analytically describes the **optical distortions**: it provides the real position of the slits on the detector.



Instrument optical distortions representation



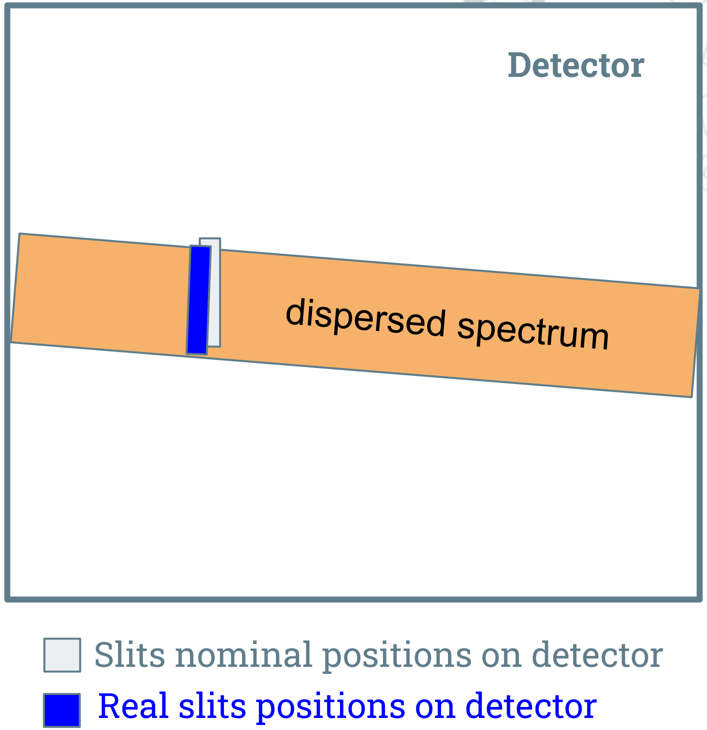
Slits nominal positions on detector

Real slits positions on detector

The three main SIPGI pillars - *the instrument model*

2.2 The spectra location model

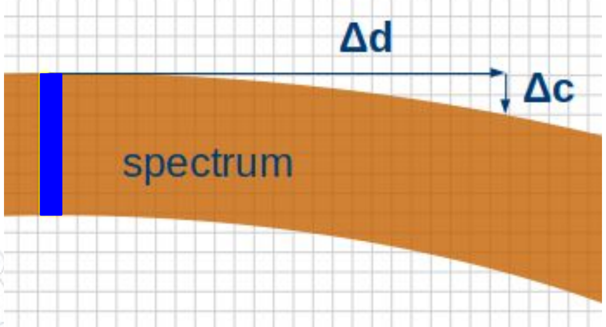
Once known the real slit position, in principle we can predict the spectra location.



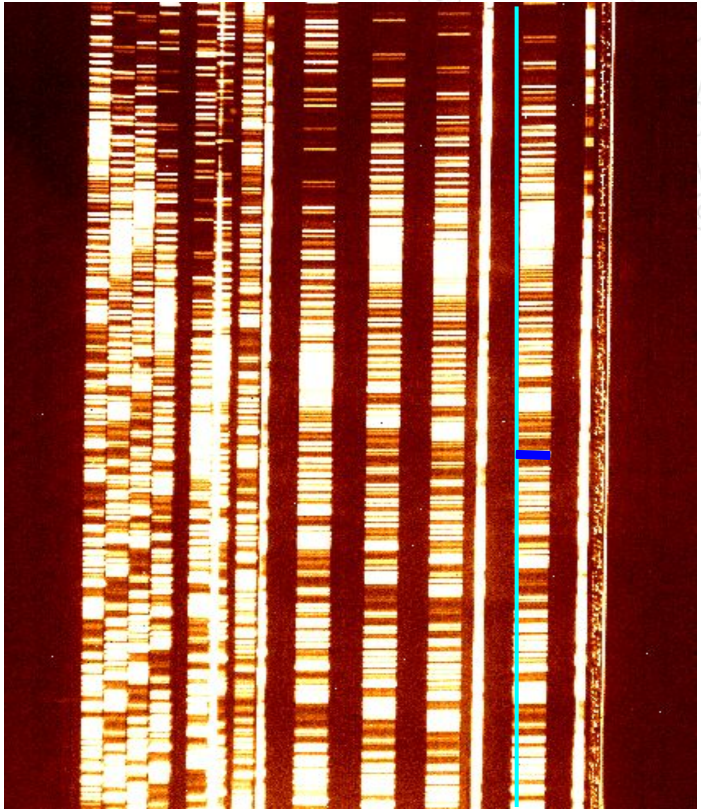
The three main SIPGI pillars - *the instrument model*

2.2 The spectra location model

The **spectra location model** analytically describes the spectra **displacement (Δc)** wrt the **ideal dispersion direction** as a function of the distance from the slit position **(Δd)**.



Spectrum curvature representation

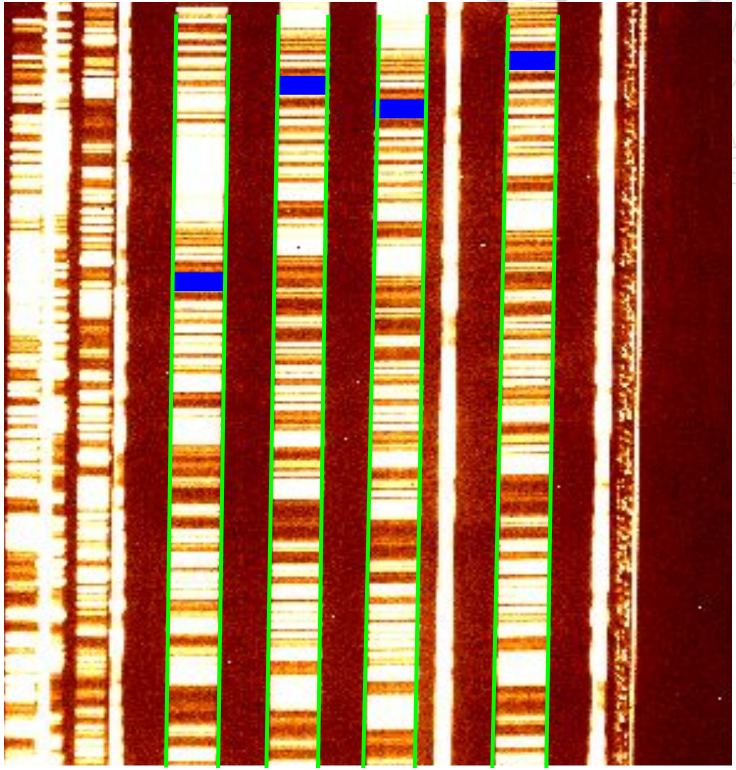


The three main SIPGI pillars - *the instrument model*

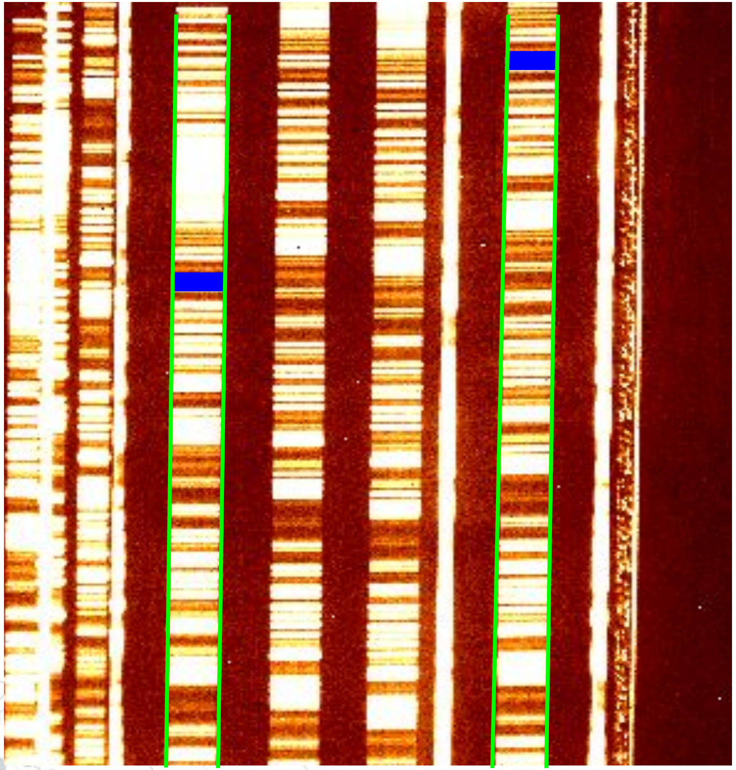
2.3 The lambda calibration model

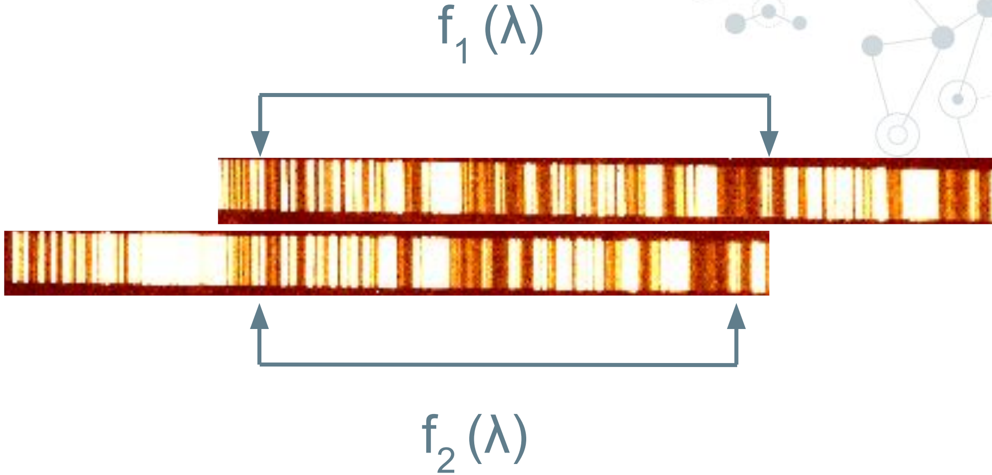
Once known the instrument configuration, the slits positions, and the spectra location, we could exactly predict the λ to be associated to each pixel of the slits.

Optical distortions affect this relation.



The three main SIPGI pillars - *the instrument model*





The wavelength solution continuously changes moving along the cross dispersion direction.

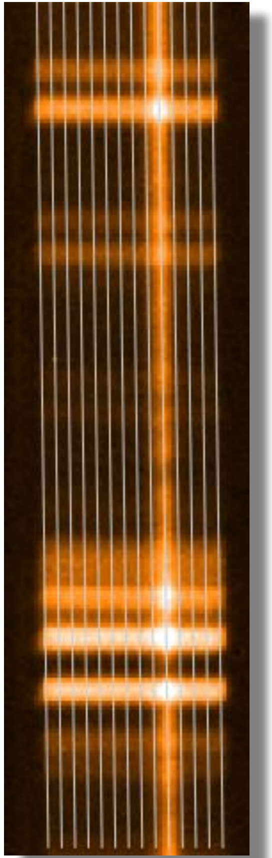
The three main SIPGI pillars - *the instrument model*

2.3 The lambda calibration model

We treat each pixel-column of each slit as a 1D spectrum.

The **lambda calibration model** analytically describes

the **λ ⟶ pixel relation**



Models are mask independent

- **Given an instrument configuration**: grism, camera, dichroic, … models are mask independent. Once the models are calibrated, if **the instrument is stable**, they **describe every mask.**

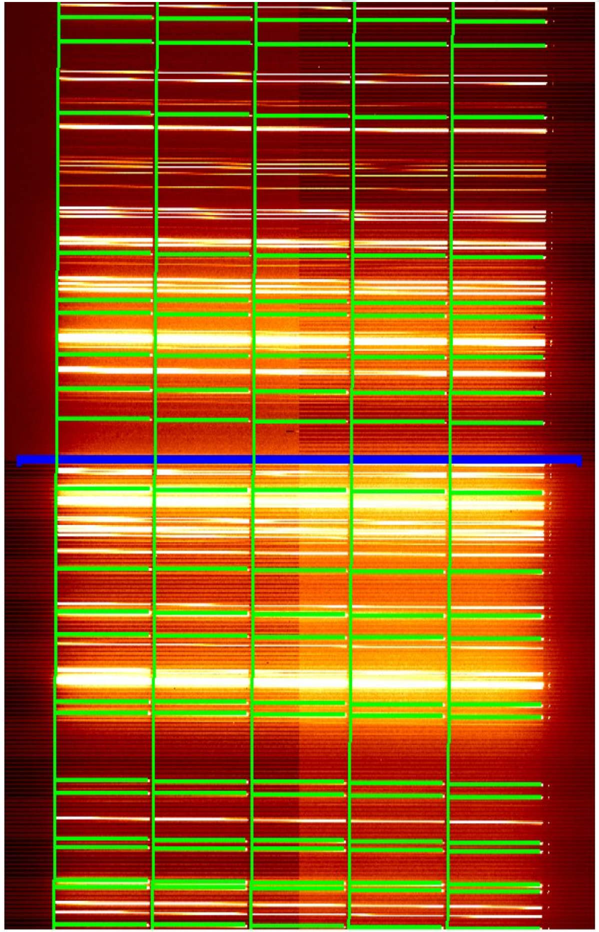
- We **calibrated** each model on **real data** for all the standard LBT-MODS/LUCI configurations.

- The SIPGI version we released includes all these instrument models.

The three main SIPGI “pillars”

Distortions can change on nightly basis.

The **models** can be **checked and adjusted** using a graphical tool.

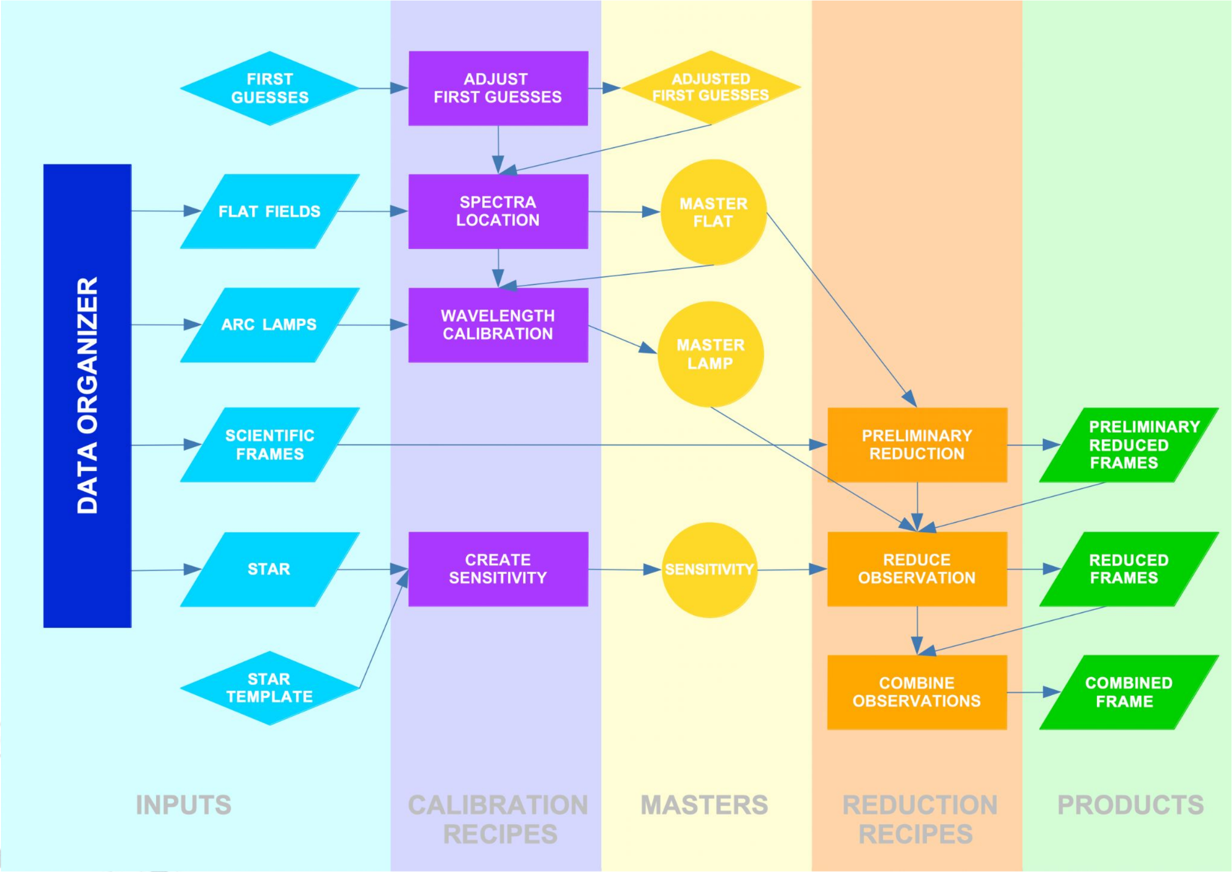


The three main SIPGI "pillars"

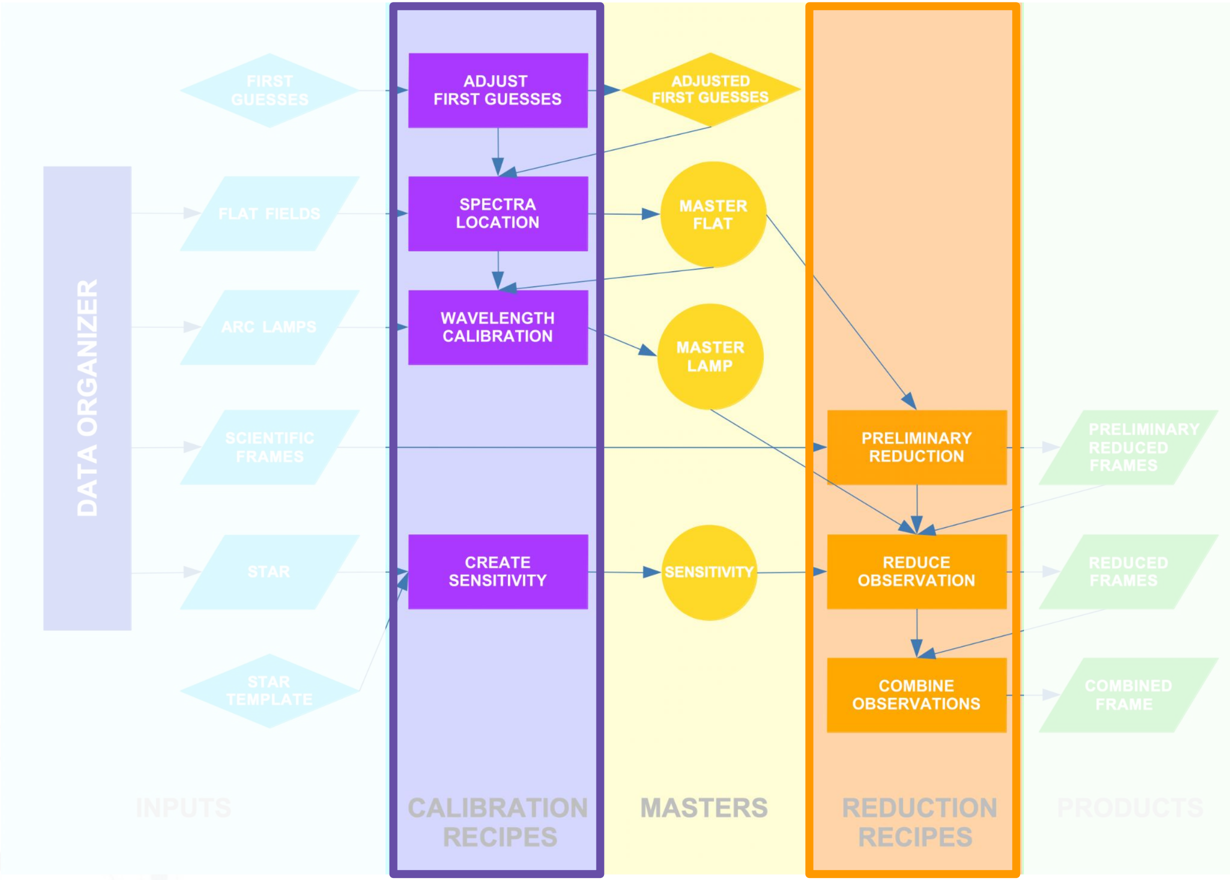
3. The recipe organization

- a complete reduction can be performed executing just 7 recipes;
- the recipes number is minimized while preserving the possibility to verify the main stages of data-reduction process with provided tools;
- the specific behaviour of each recipe is controlled through a set of input parameters set by the user.

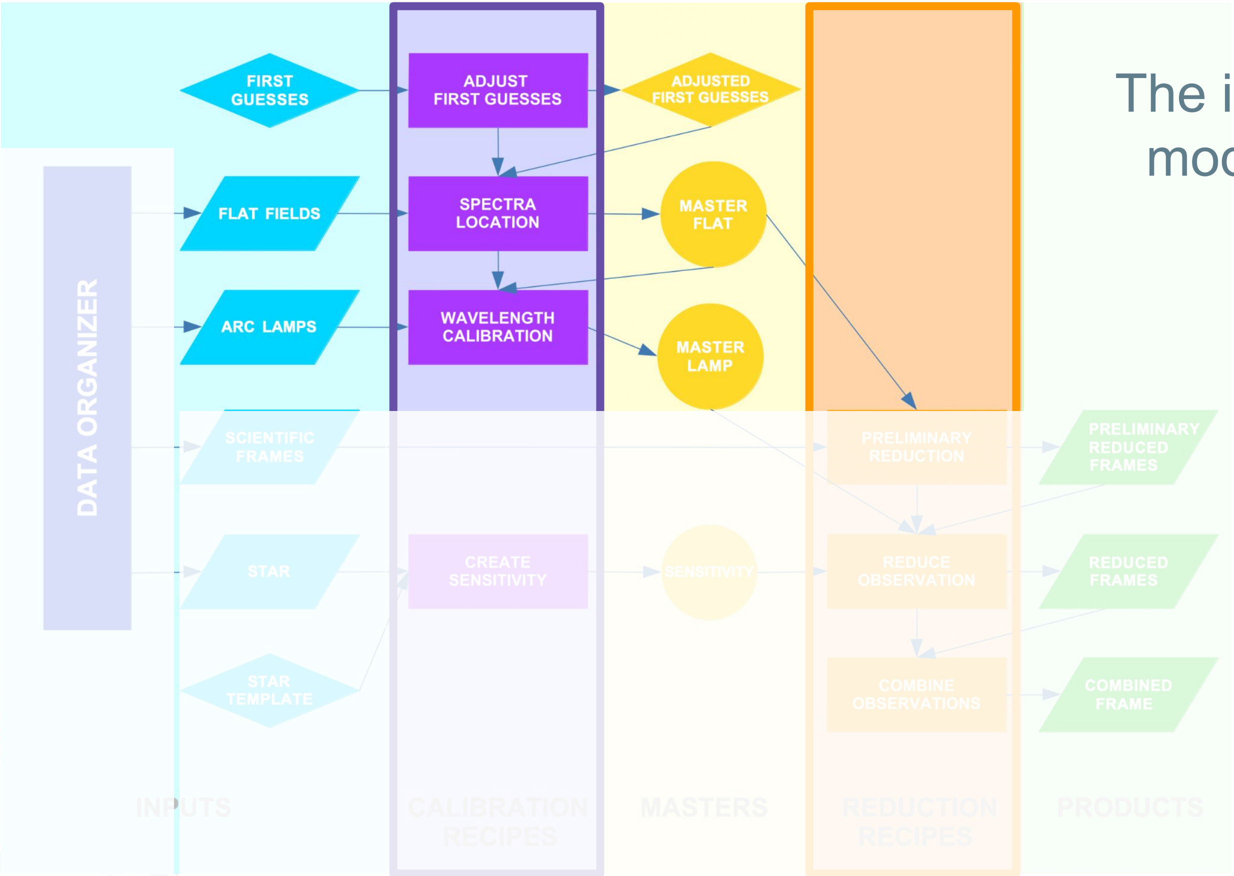
The logical scheme of SIPGI functioning



The logical scheme of SIPGI functioning

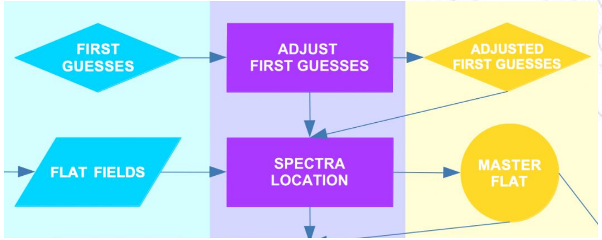


The logical scheme of SIPGI functioning



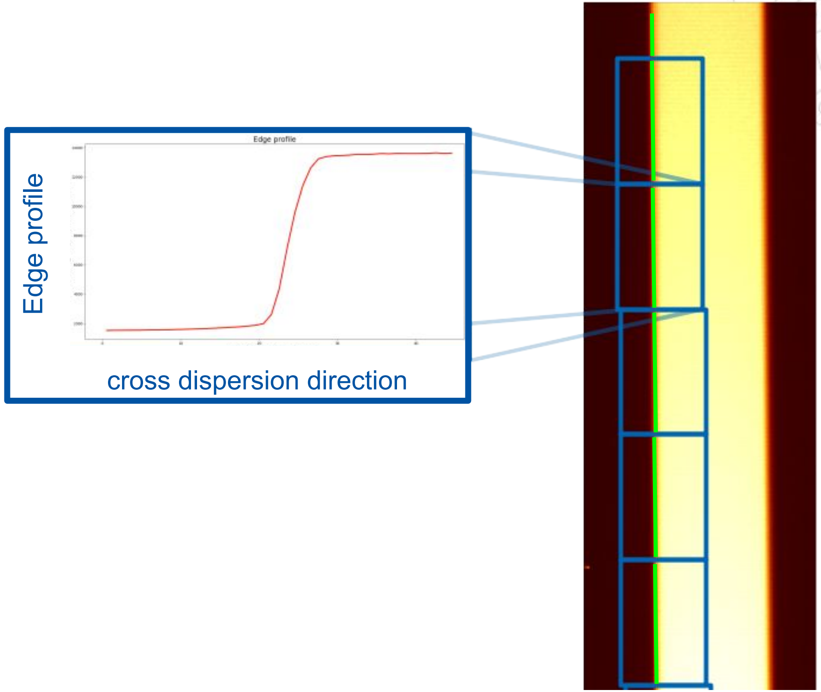
The instrument model realm

Spectra tracing refinement (Create Master Flat)

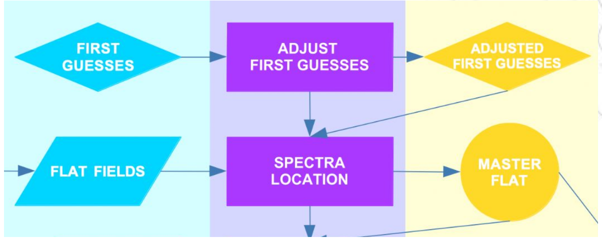


The Create Master Flat recipe refines the spectral tracing for each slit.

- It uses the **adjusted first guesses** of the spectra location model to locate the spectrum position on flat frame;
- It cuts thumbnails along the edge and collapses them in the dispersion direction to create the edge profile;
- It computes the numerical derivative of the profile and identifies the edge position as the point of maximum curvature;
- It fits the edge positions to define the **real spectrum trace**;

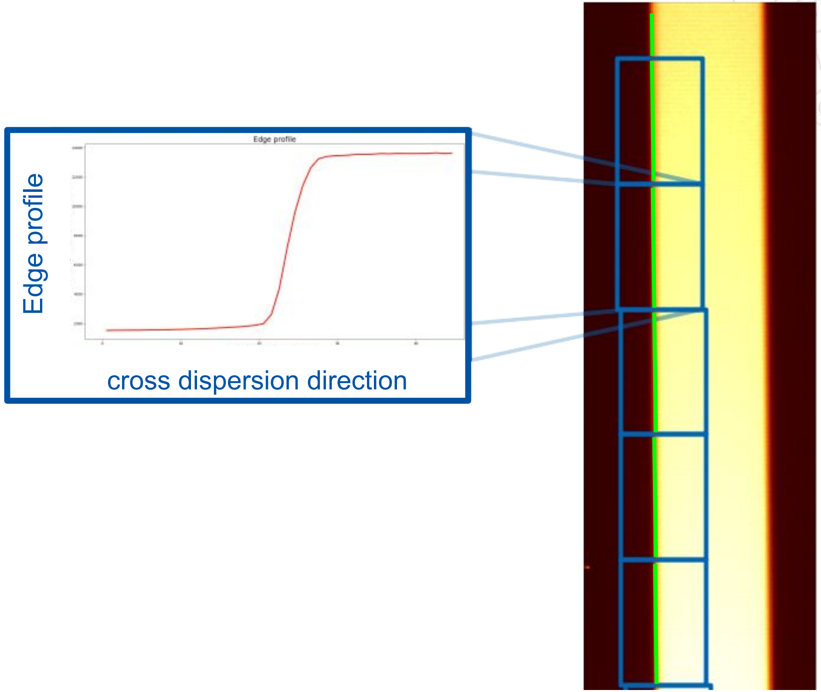


Spectra tracing refinement (Create Master Flat)

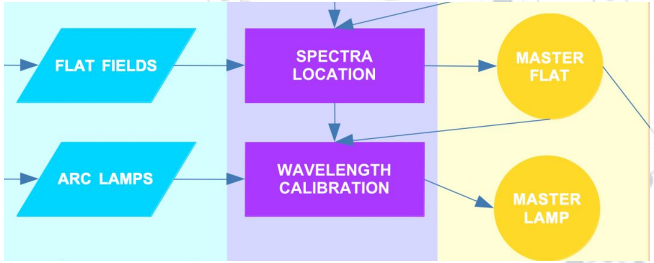


Output: Master Flat (first calibrator)

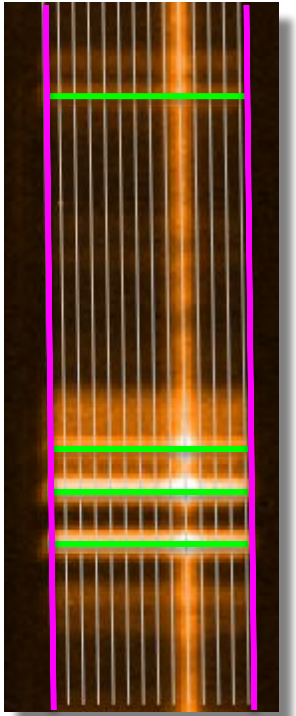
- spectra location;
- px-to-px variation;
- the adjusted first guesses of the lambda calibration model.



Lambda calibration refinement (Create Master Lamp)

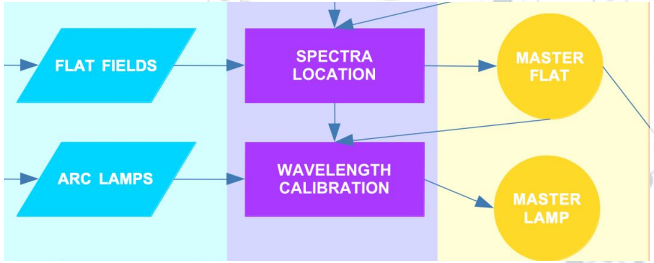


The Create Master Lamp recipe refines the wavelength solution for each pixel-column of each slit

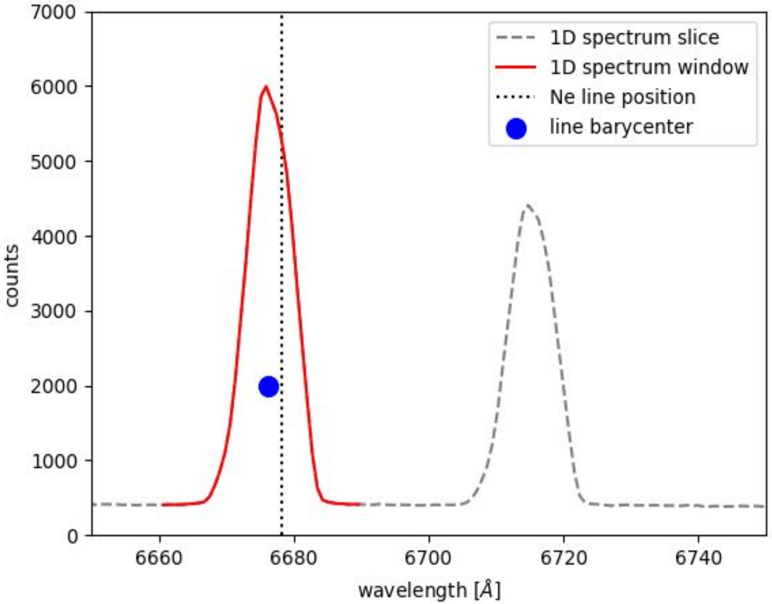


- It applies the adjusted lambda calibration model to locate the lines;
- It slices the 2D spectra in 1-pixel spectra;

Lambda calibration refinement (Create Master Lamp)

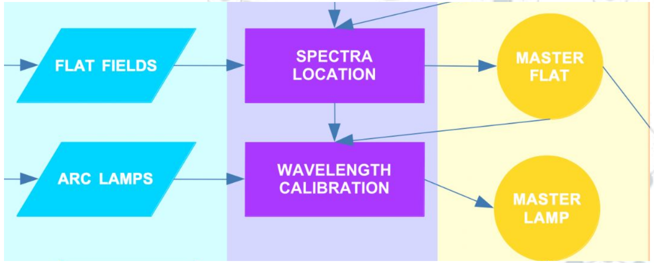


The Create Master Lamp recipe refines the wavelength solution for each pixel-column of each slit

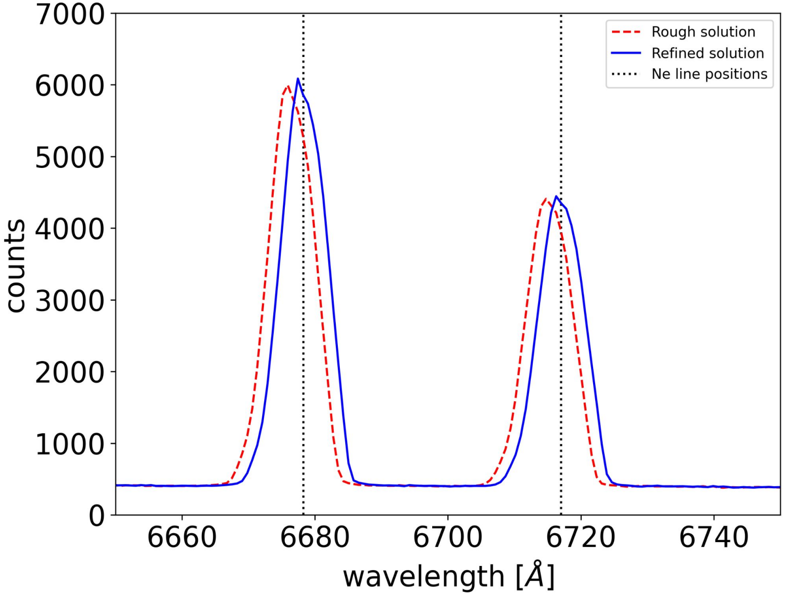


- It applies the adjusted lambda calibration model to locate the lines;
- It slices the 2D spectra in 1-pixel spectra;
- It lambda-calibrates the 1d-spectra applying the adjusted model;
- It measures the **real line position** as the barycenter of a spectrum region set by the user;

Lambda calibration refinement (Create Master Lamp)

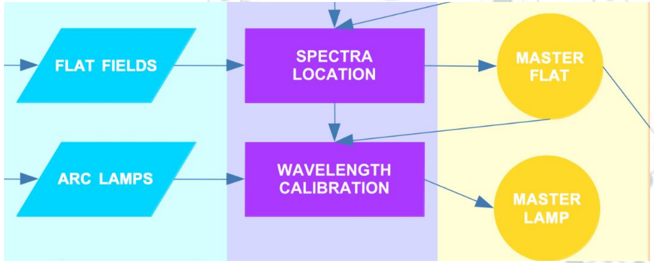


The Create Master Lamp recipe refines the wavelength solution for each pixel-column of each slit



- It applies the adjusted lambda calibration model to locate the lines;
- It slices the 2D spectra in 1-pixel spectra;
- It lambda-calibrates the 1d-spectra applying the adjusted model;
- It measures the **real line position** as the barycenter of a spectrum region set by the user;
- It fits the real lines positions (i.e. all the ●) to defines the real lambda-calibration relation.

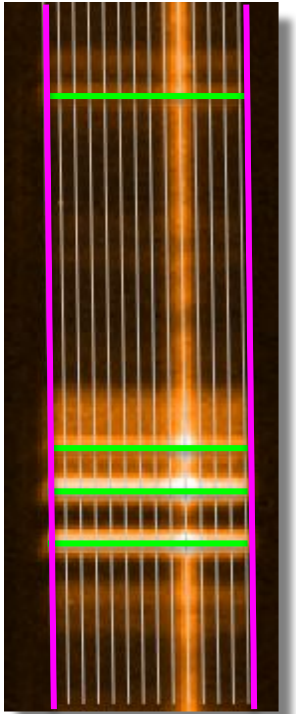
Lambda calibration refinement (Create Master Lamp)



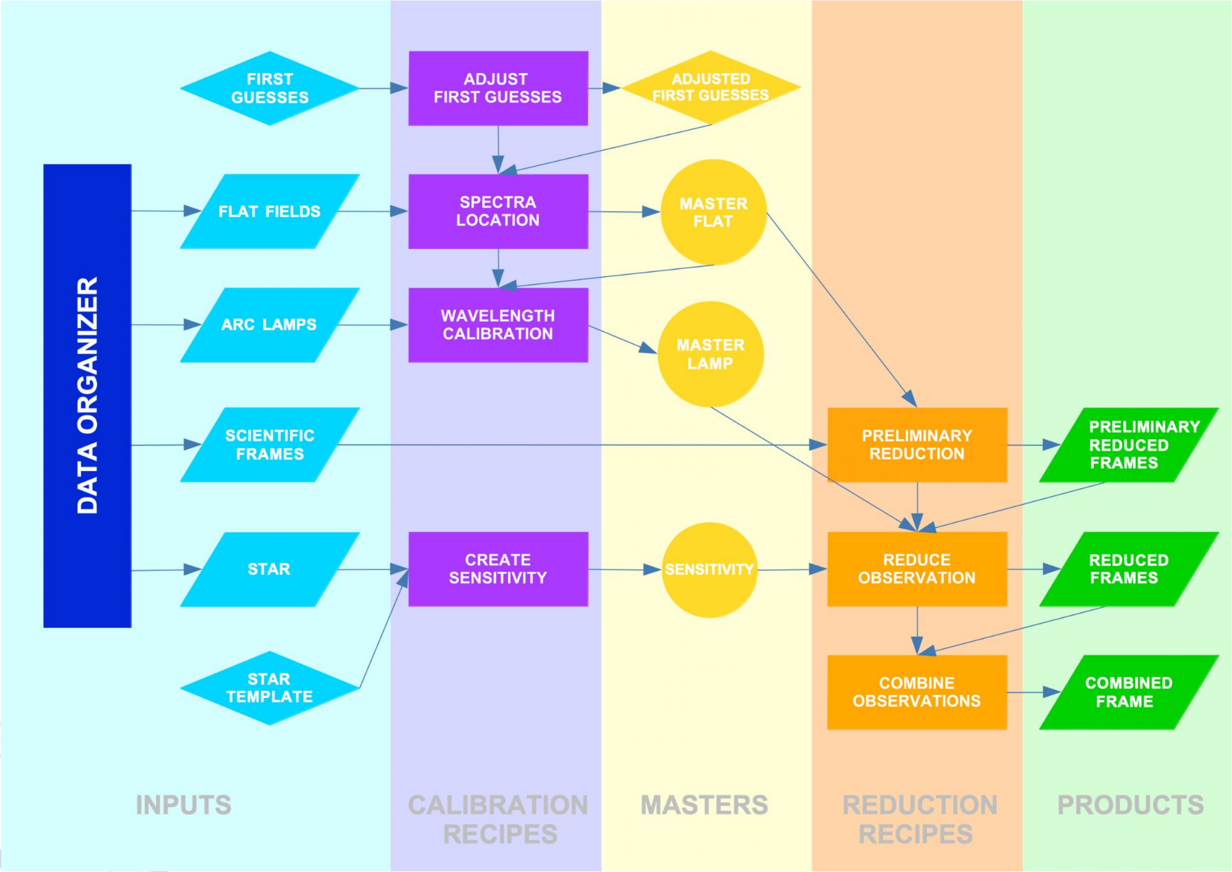
Output: Master Lamp (second calibrator)

For each slice of each slit, user has now
a model that describes the wavelength solution

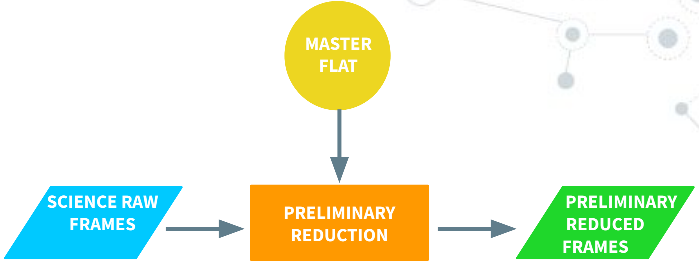
$$m_{slit}(n_{slice}, \lambda)$$



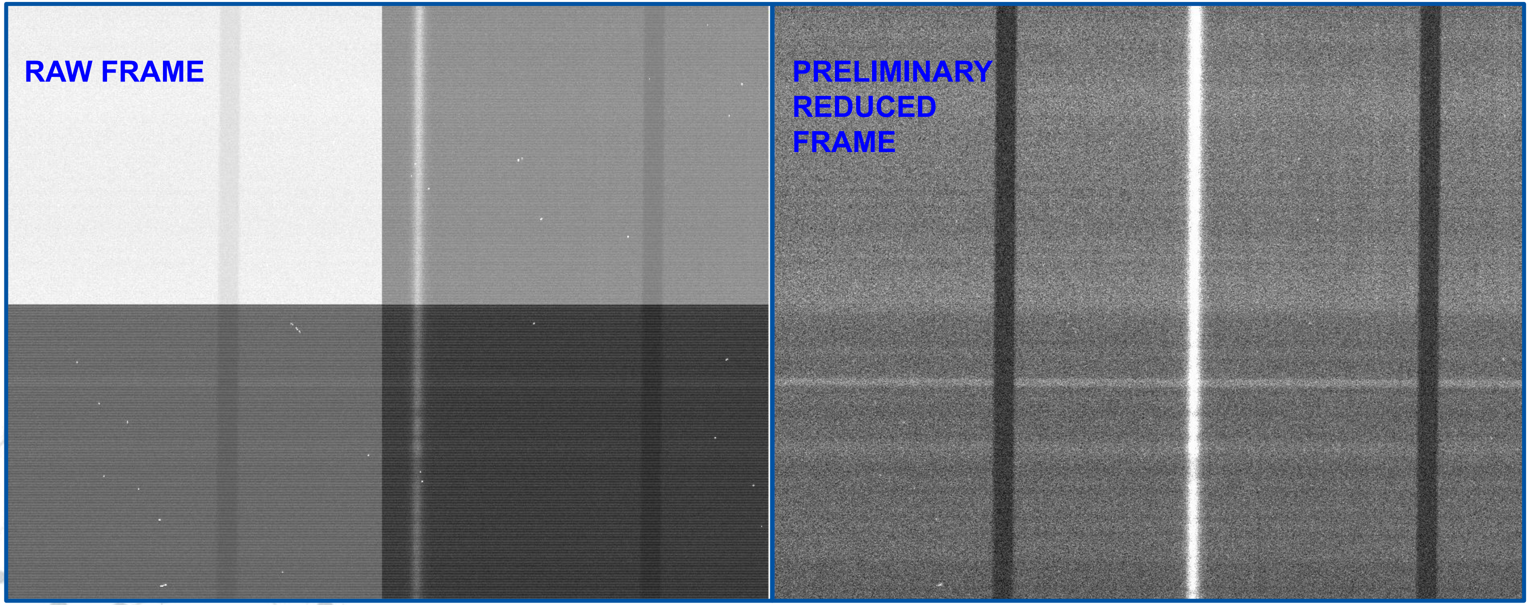
The logical scheme of SIPGI functioning



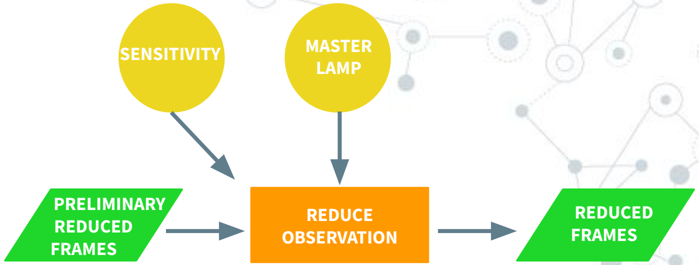
The preliminary reduction



The recipe: ***1.*** subtracts the bias/dark level; ***2.*** corrects for px-to-px variation; ***3.*** corrects bad pixels and cosmic rays.



Reduce Observation



The recipe:

- **wavelength calibrates** each spectrum and extracts the 2D spectrum;
- **detects objects**;
- **extracts 1D spectrum**.

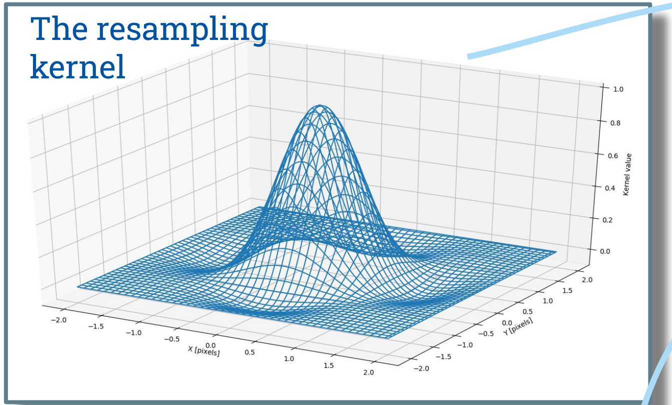
Optionally:

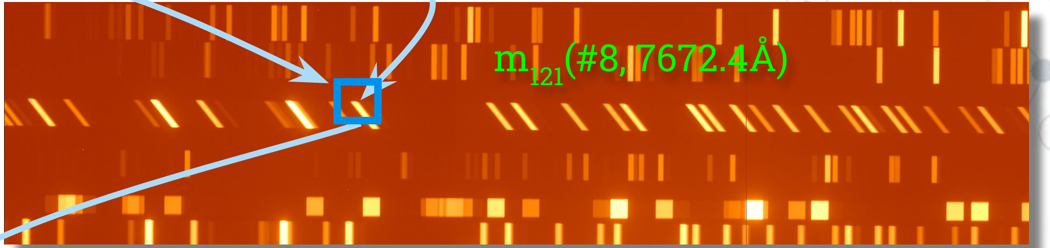
- **performs the sky subtraction** (both on wavelength-calibrated and not calibrated frames). It can estimates the background level:
  - from the frame itself;
  - from dithered frames, e.g ABBA/Davies (Davies only for LUCI wavelength calibrated frames);
- **flux calibrates 1D spectra** (and possibly also 2D spectra)

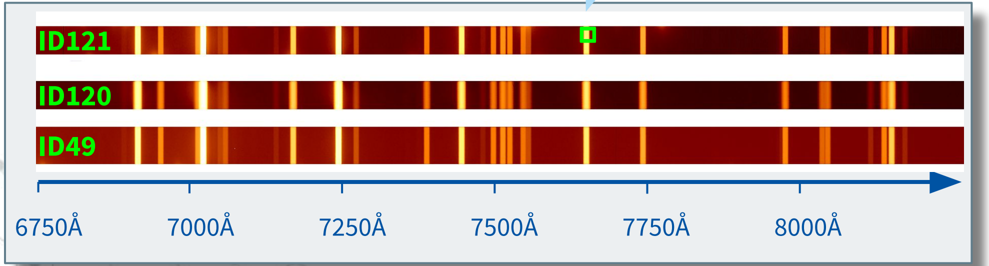
2D extraction

Models: $m_{slit}(n_{slice}, \lambda) \rightarrow$ pixel

$\lambda \in [\lambda_{min}, \lambda_{max}]$
$\Delta\lambda = \lambda_{n+1} - \lambda_n$



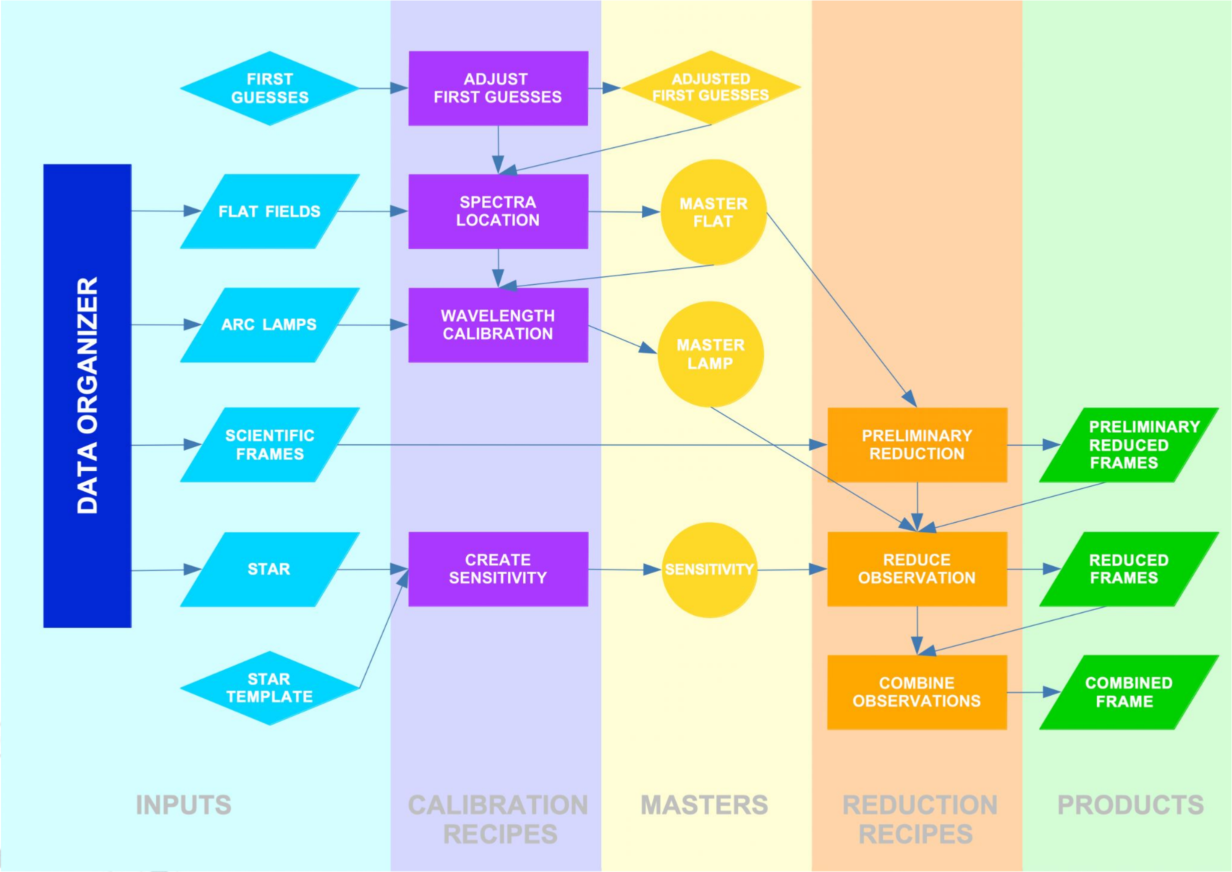




SIPGI:

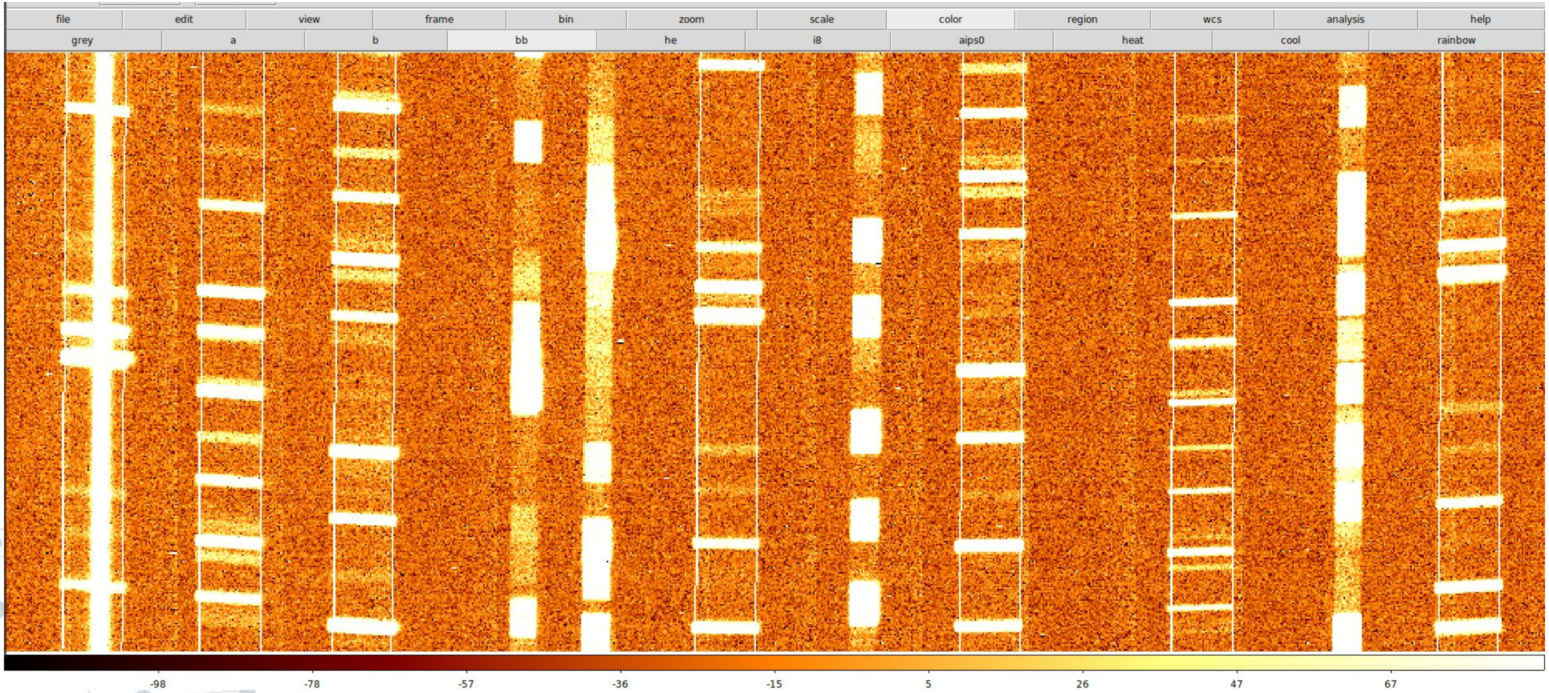
1. ingests the λ-range and Δλ;
2. for each slit, uses its **refined models** (m_{slit}) to locate wavelengths (of the current slice) on the frame;
3. applies the 2D **resampling kernel;**
4. extracts 2D spectra **λ-calibrated** and **corrected by distortions.**

The logical scheme of SIPGI functioning



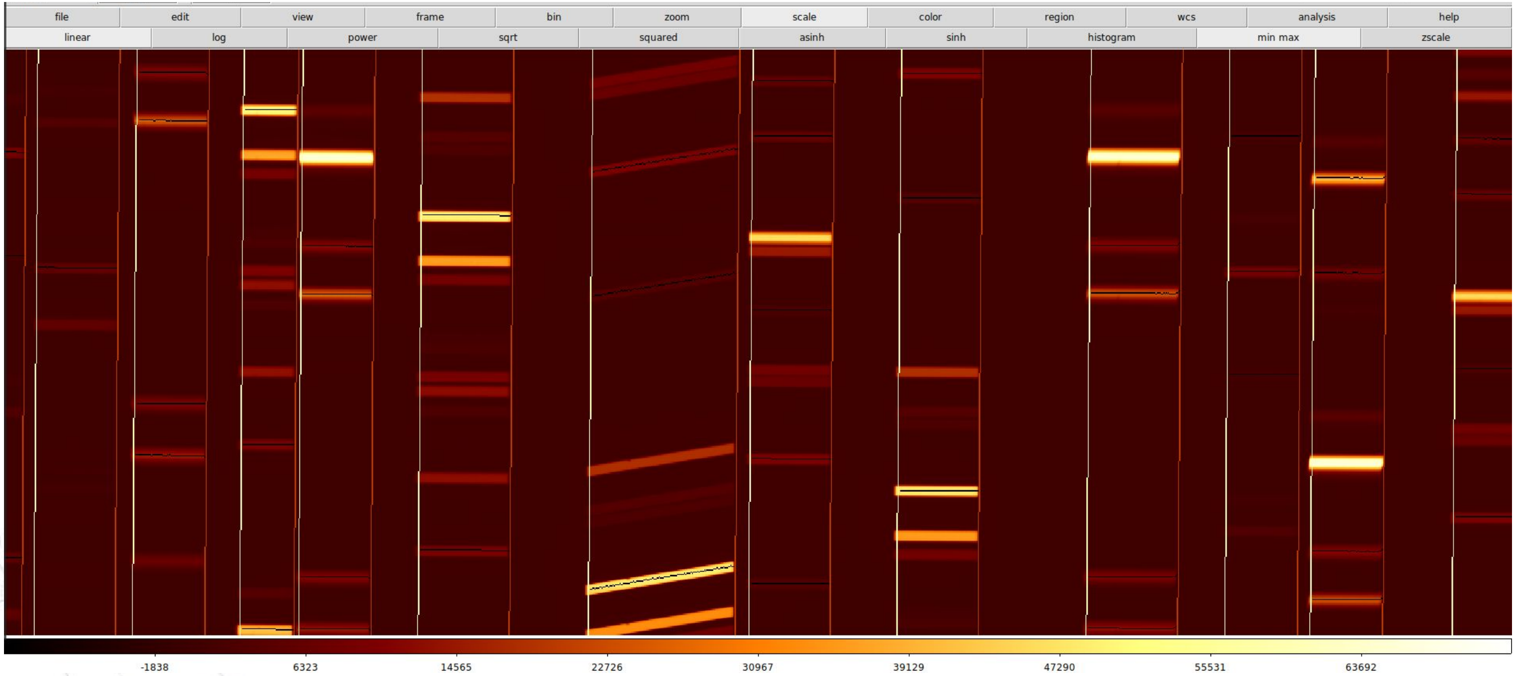
The quality control tools - the first calibrator: MsFlat

- **Show Spectra Location:** it allows to check the quality of MsFlat



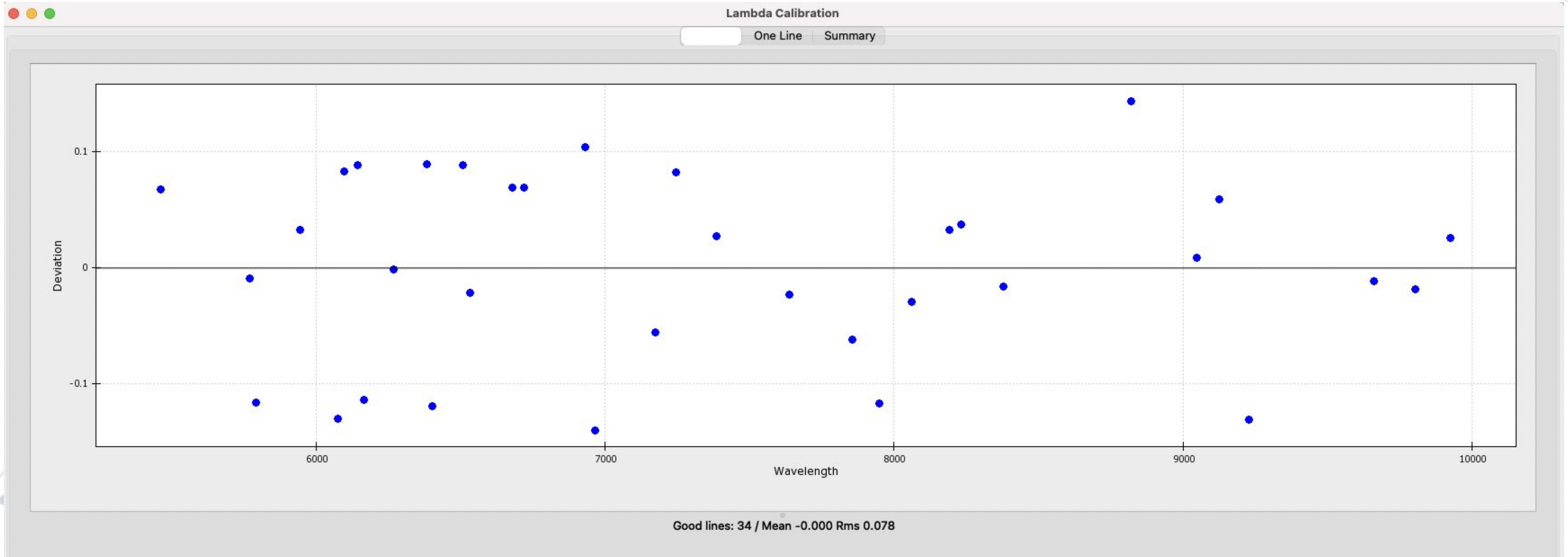
The quality control tools - the second calibrator: MsLamp

- **Show Lambda Cal:** it allows to *qualitatively* check the MsLamp



The quality control tools - the second calibrator: MsLamp

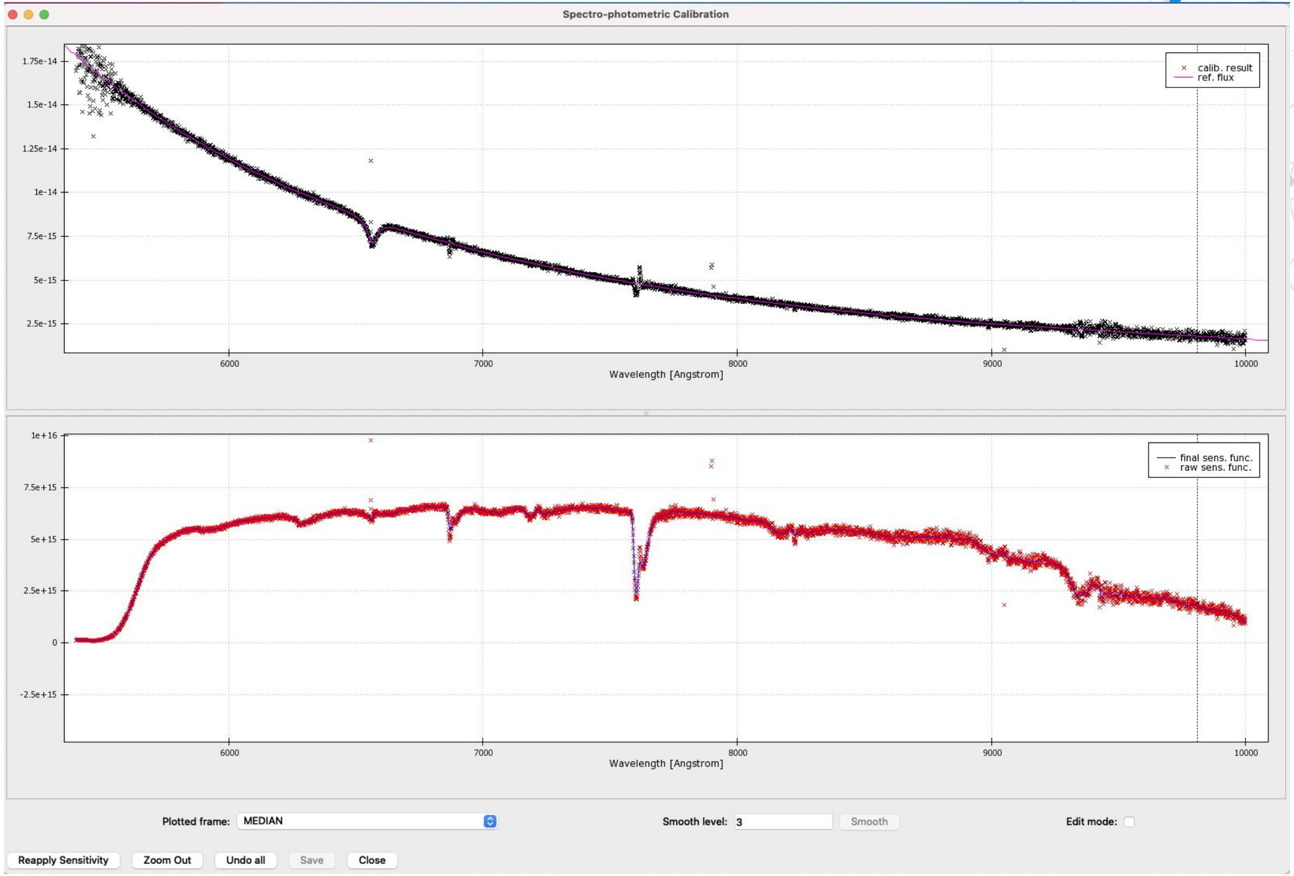
- **Check Lambda Cal & Plot Lambda Cal:** they allow to *quantitatively* check the MsLamp



The quality control tools - the third calibrator: Sensitivity

- **Plot Sensitivity:**

it allows to check the Sensitivity Function



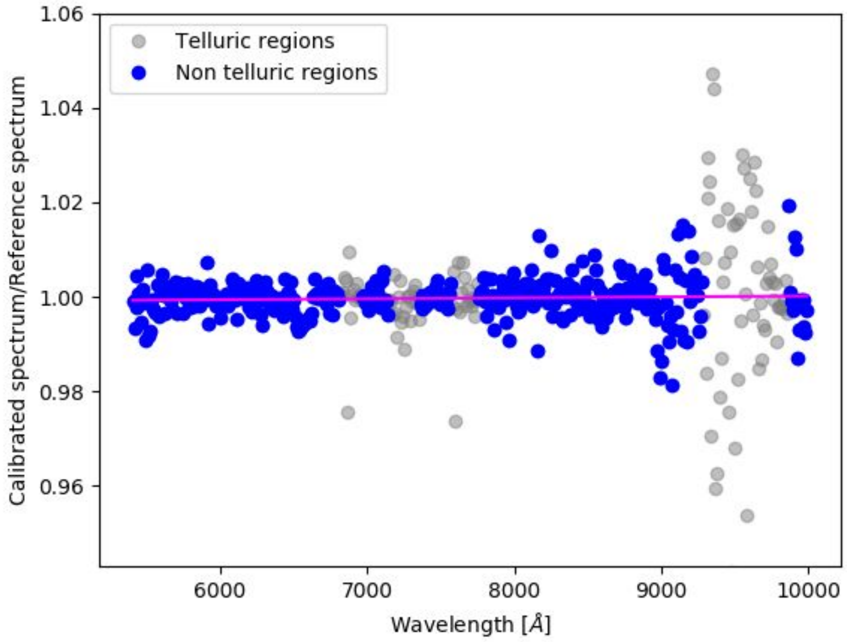
The SIPGI performances - the wavelength calibration

- **Wavelength calibration accuracy better than ⅕ px** in 90% of the cases;
- It does not depend on the position of the slit within the FoV;

The SIPGI performances - the flux calibration

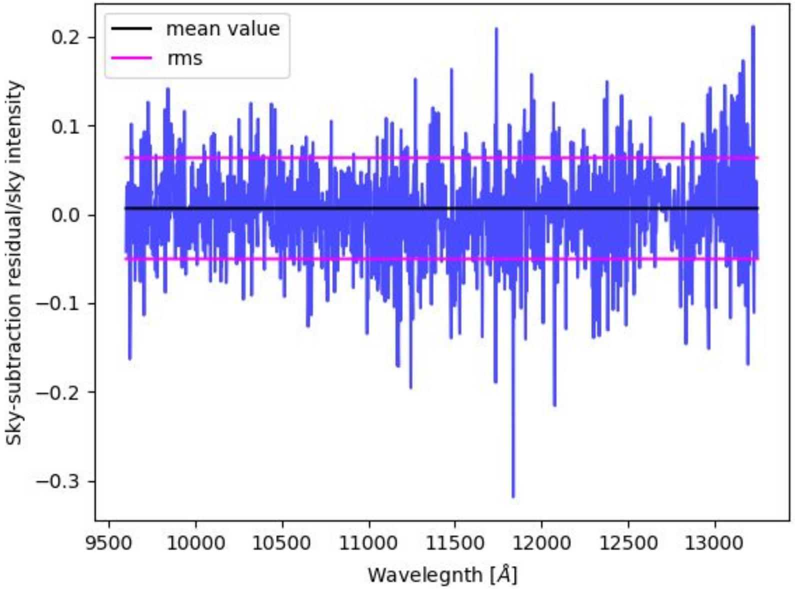
- The typical sensitivity function is able to **fully recover the shape of the spectrum**;

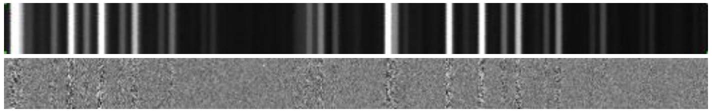
- The typical rms in the flux calibration is 0.4%(0.5%) in the regions not affected by telluric absorption for MODS(LUCI) data and 2%(5%) in regions affected by telluric absorptions.



The SIPGI performances - the sky subtraction

- The mean value of sky-subtraction residuals relative to the sky intensity as a function of the wavelength for a typical LUCI observation with R = 1000 is 0.006 with a rms of ~6%.





The SIPGI performances

Belladitta et al. 2020

Mignoli et al. 2020

Polletta et al. 2021

Saracco et al. 2021

From SIPGI to SpectraPy

- We are now working to extend SIPGI to other optical/near-IR through-slit spectrographs;

- However, the interaction we had with PIs during these 10 years has highlighted that the most “demanding” task is to obtain the **2D spectra lambda calibrated and free of distortions**;

    - **This has led to the development of SpectraPy.**

SpectraPy (by Marco Fumana)

SpectraPy is a *spectrograph independent* Python library

focused on the **extraction of 2D wavelength calibrated spectra.**

It is not a standalone software (differently from SIPGI).

It inherits the SIPGI **instrument model concept**, and allows to apply it to data acquired with all the through-slit spectrographs.

Differently from SIPGI, it does not provide the instrument model,

but **allows PIs to derive them.**

SpectraPy (by Marco Fumana)

1. **Instrument configuration file**
   - Detectors description
   - Grism description
   - File description
   - Mask

```
[Description]
instrument = MODS1R/2R
grism = G670L

[Detector]
pixel_scale = 0.123
pixel_size = 0.015
xpixels = 8288
ypixels = 3088

[Grism]
dispersion_direction = RL
linear_dispersion = 0.8 #A/pixels
reference_lambda = 6929.47 #Ne line

[Files]
data_hdu = Primary
```

2. **Mask description file**

   Required to automatically locate spectra on the frames

#ID	DIMX	DIMY	X	Y	ROT	WID	LEN	REF
49	0.720	3.600	-73.62	-86.11	0.0	1.2	6.0	0
85	0.720	6.000	-40.32	-91.09	0.0	1.2	10.0	0
119	0.720	3.600	-86.62	-48.47	40.0	1.2	6.0	0
Ref1	2.400	2.400	-20.70	74.29	0.0	4.0	4.0	1

SpectraPy models calibration workflow

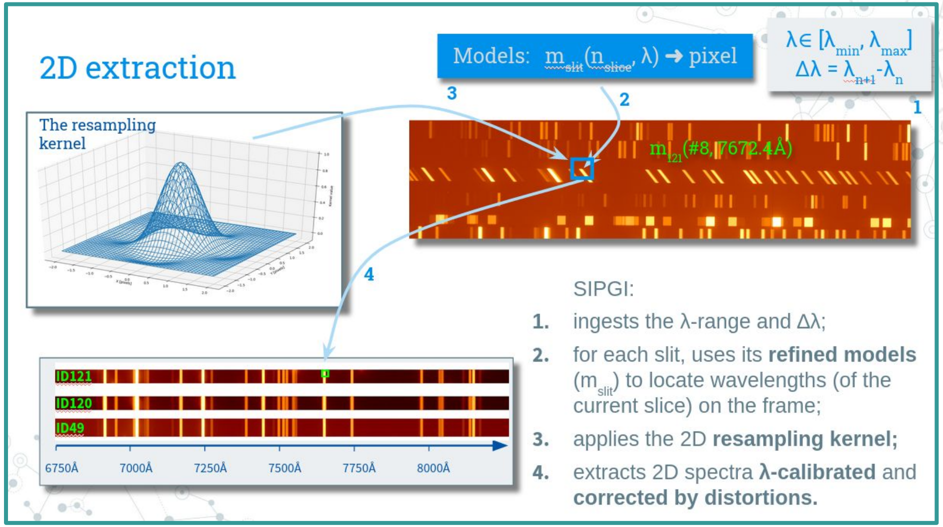
Calibration workflow is the same for each model:

- SpectraPy displays data as **DS9 frames,** and it visualizes models **creating DS9 regions** on expected positions;
- User **manually adjusts** the location of these regions;
- SpectraPy **refits** these new positions, **updating** the **global model solution**

SpectraPy 2D extraction

SpectraPy follows the same SIPGI 2D extraction scheme.

- It provides 2D spectra wavelength calibrated.
- It does not take into account:
  - bias/dark subtraction;
  - bad pixels/cosmic rays correction;
  - px-to-px variation;
  - sky subtraction;
  - flux calibration;
  - 1D extraction.



Why this choice?

Astropy provides packages for spectra analysis (specutils, linetools, pyspeckit, …), but spectra reduction package is *missing*.

SpectraPy wants to be the 1st brick of this missing package.

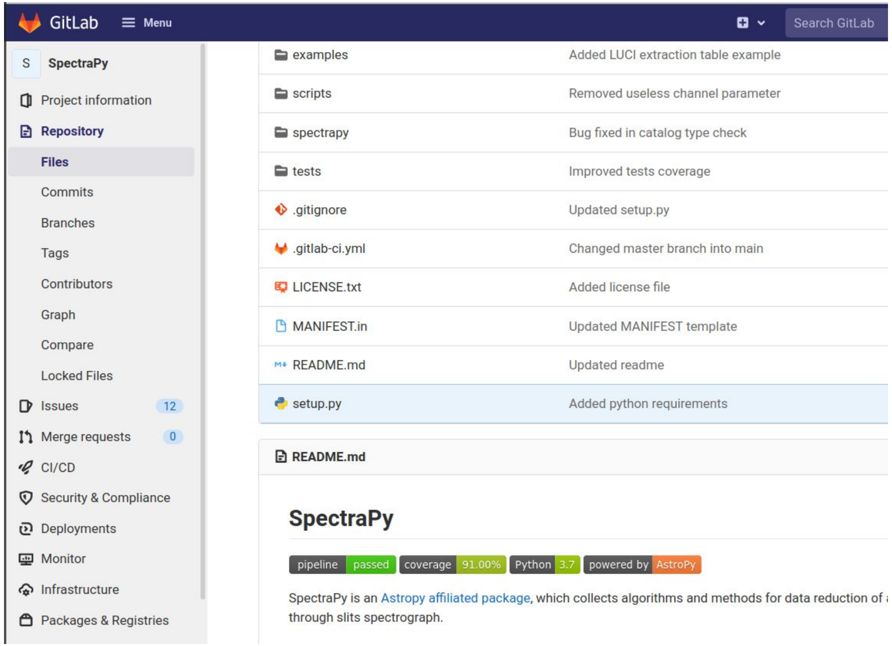
SIPGI download

- Pandora page
  http://pandora.lambrate.inaf.it/sipgi/

- SIPGI DOI:
  10.20371/inaf/sw/2021_00002

SpectraPy download

- SpectraPy is an **Astropy affiliated package**
  https://www.astropy.org/affiliated/
- Open gitlab repository
  https://gitlab.com/mcfuman/SpectraPy/
- Pandora page
  http://pandora.lambrate.inaf.it/SpectraPy/
- Documentation
  https://mcfuman.gitlab.io/SpectraPy/
- SpectraPy DOI:
  10.20371/inaf/sw/2021_00001



Thank you for your attention.

For those interested, after the seminar,
Susanna Bisogni
holds a practical tutorial on using SIPGI.
The tutorial will be uploaded at

http://pandora.lambrate.inaf.it/sipgi/