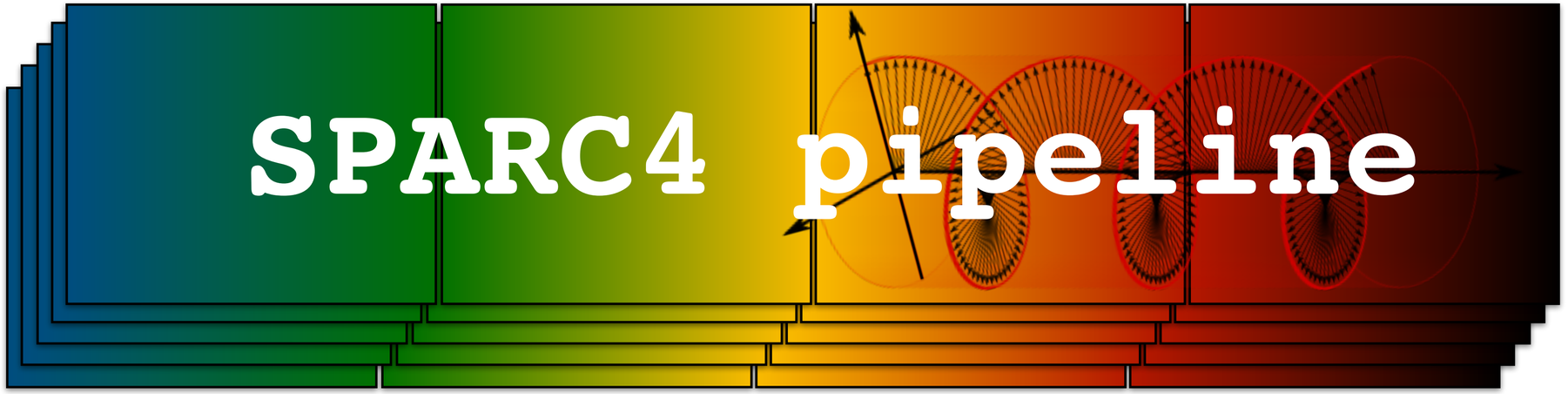
SPARC4 Pipeline: Hands-On Data Reduction for Multi-Band Photometry and Polarimetry with SPARC4



Equipe do pipeline da SPARC4:

Eder Martioli[1], Cláudia V. Rodrigues[2], Júlio Campagnolo[3], Ana Carolina Mattiuci[2], Wagner Schlindwein[2], Fernando Falkenberg Marques[2], Francisco Jablonski[2], Isabel Lima[4], Marina Mello[2] , Leandro de Almeida[1], Luciano Fraga[1]

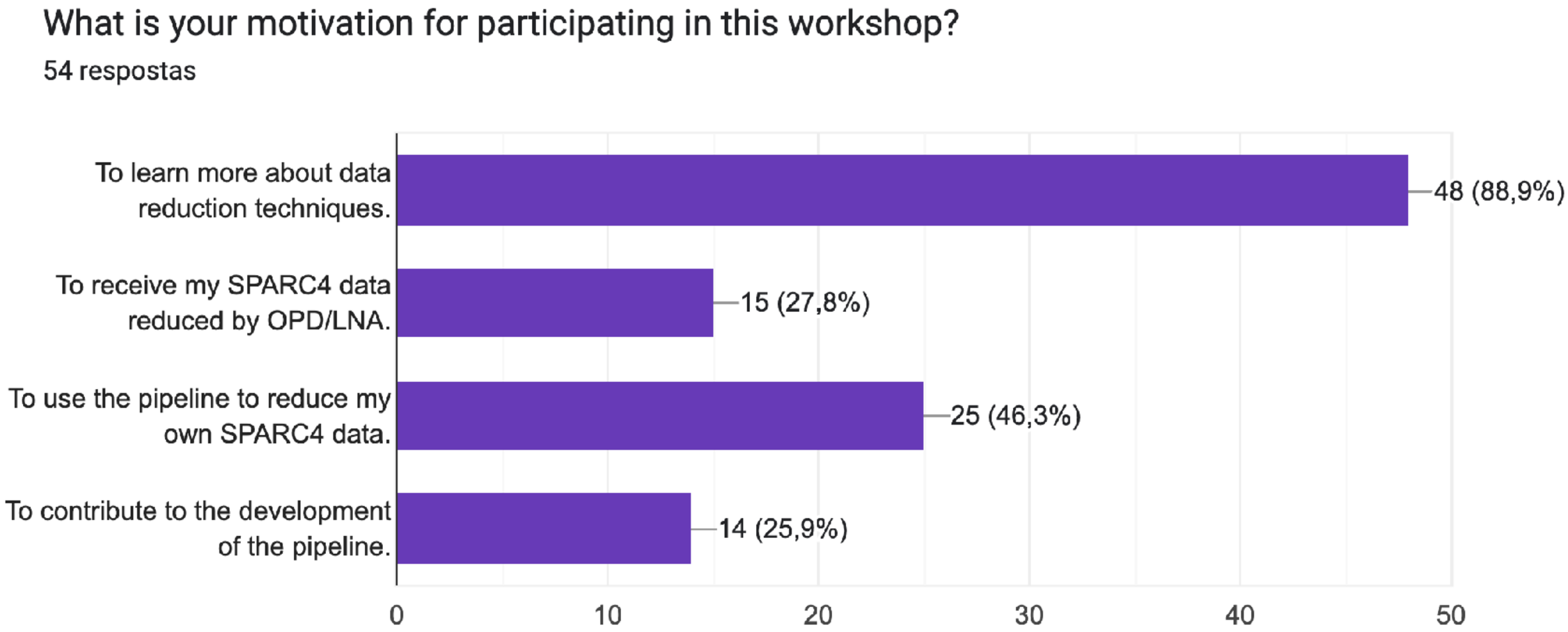
[1] LNA, [2] INPE, [3] CEFET/RJ, [4] UNESP

XLVII Reunião Anual da SAB, Águas de Lindoia, 26 de setembro de 2024

PROGRAM	
15:00 - 15:30	Introduction to and installation of the SPARC4 Pipeline.
15:30 - 16:00	Preparatory steps: the structure of raw data and parameter files.
16:00 - 16:20	Reduction of the minidata.
16:20 - 16:30	Break.
16:30 - 17:00	Products: calibrations, reduced images and stellar catalogs.
17:00 - 17:20	Products: light curve.
17:20 - 17:40	Products: polarimetry.
17:40 - 18:00	Products: polarimetric time series and polarization map.

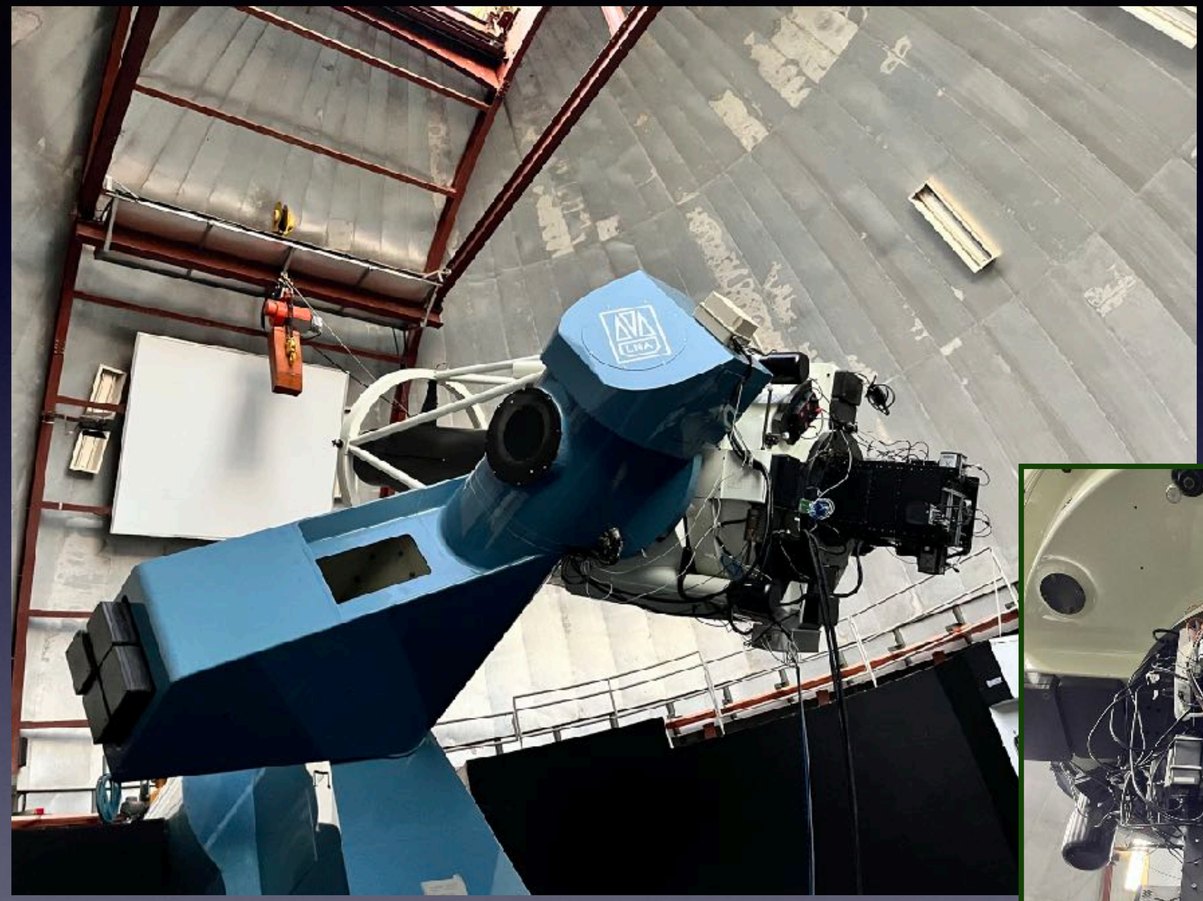
1. Introduction and Installation

You’re here for …

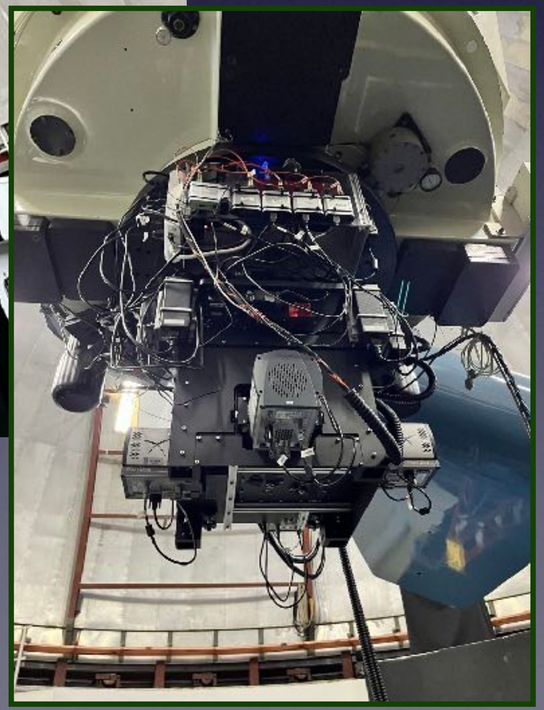


SPARC4 at Pico dos Dias Observatory

✓ 4-band imager and polarimeter



Telescope Perkin-Elmer ⌀ 1.6 m



SPARC4 at Pico dos Dias Observatory

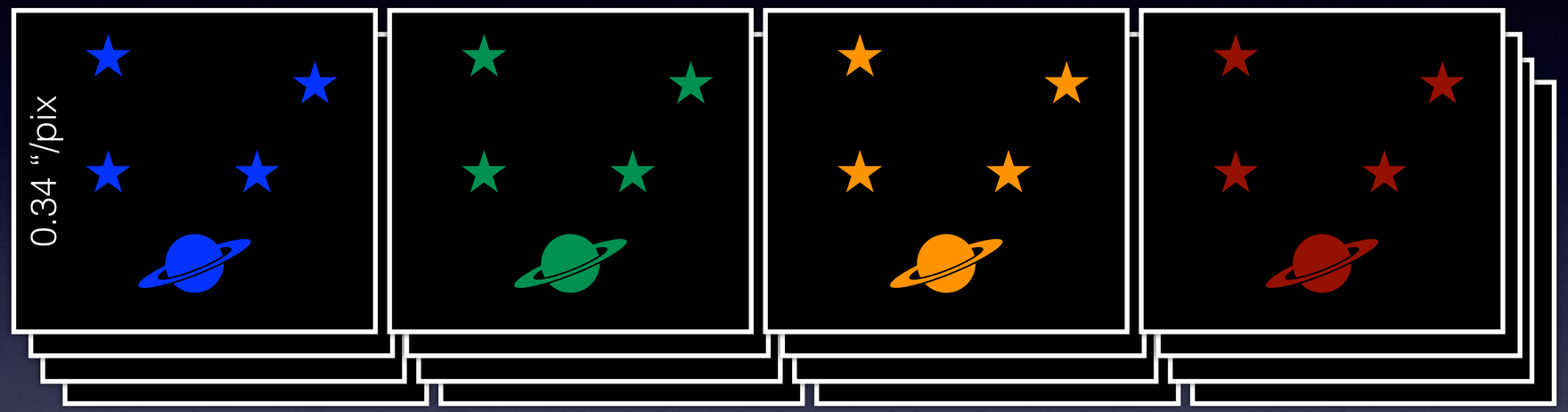
✓ 4-band imager and polarimeter

SDSS-*g'* **SDSS-*r'*** **SDSS-*i'*** **SDSS-*z'***

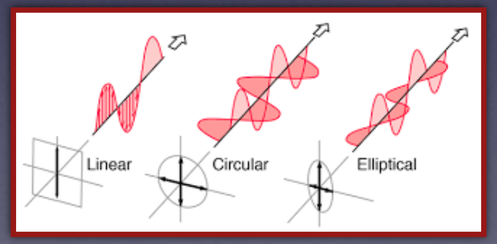
PHOT

5.6 arcmin

0.34 “/pix

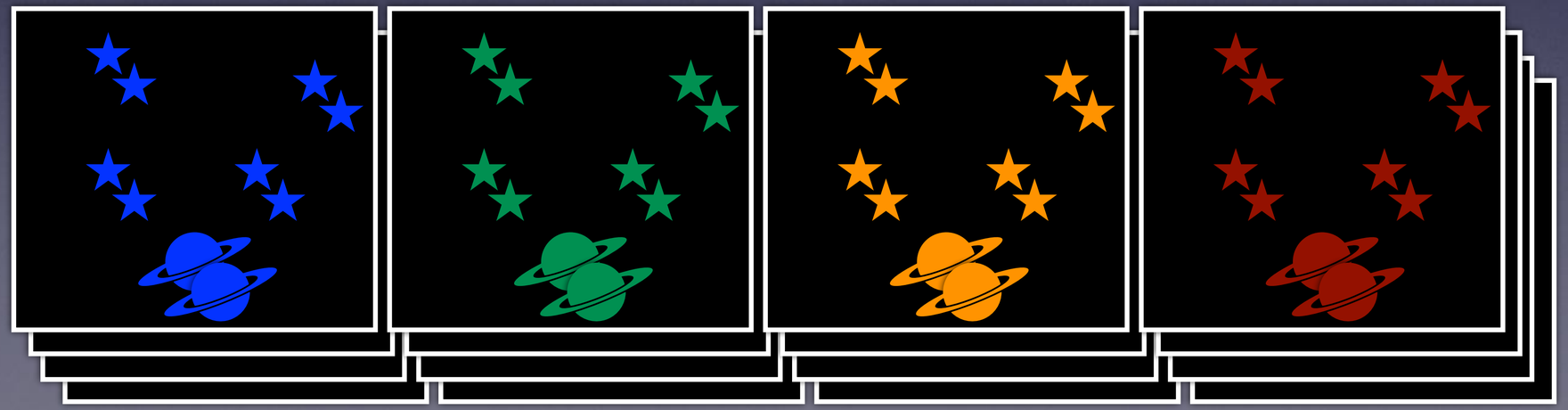


26 FPS



POLAR

$\lambda/2$ or $\lambda/4$



Pipeline concept

- The SPARC4 Pipeline is a suite of Python routines designed to automate the reduction and instrumental calibration processes for SPARC4 data collected in specific, pre-defined observation modes.

- External libraries used by the SPARC4 Pipeline:

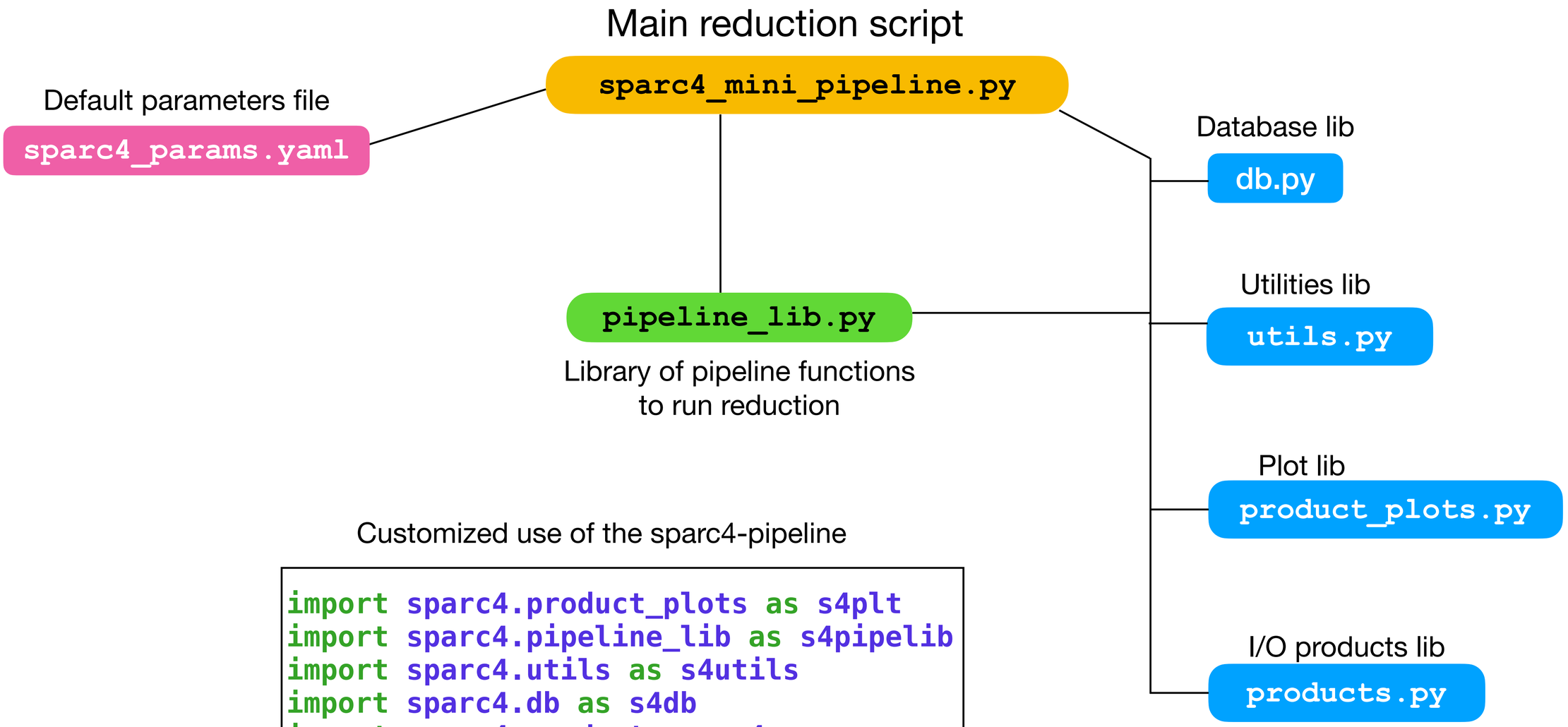
  - **`ASTROPOP`**

  - **`ASTROPY`**

  - **`NUMPY, SCIPY, ETC. (ANACONDA)`**

- Open source and available to everyone, but its main purpose is to run at LNA. Unfortunately, we cannot currently provide external user support; data reduction quality is of PI's responsibility.

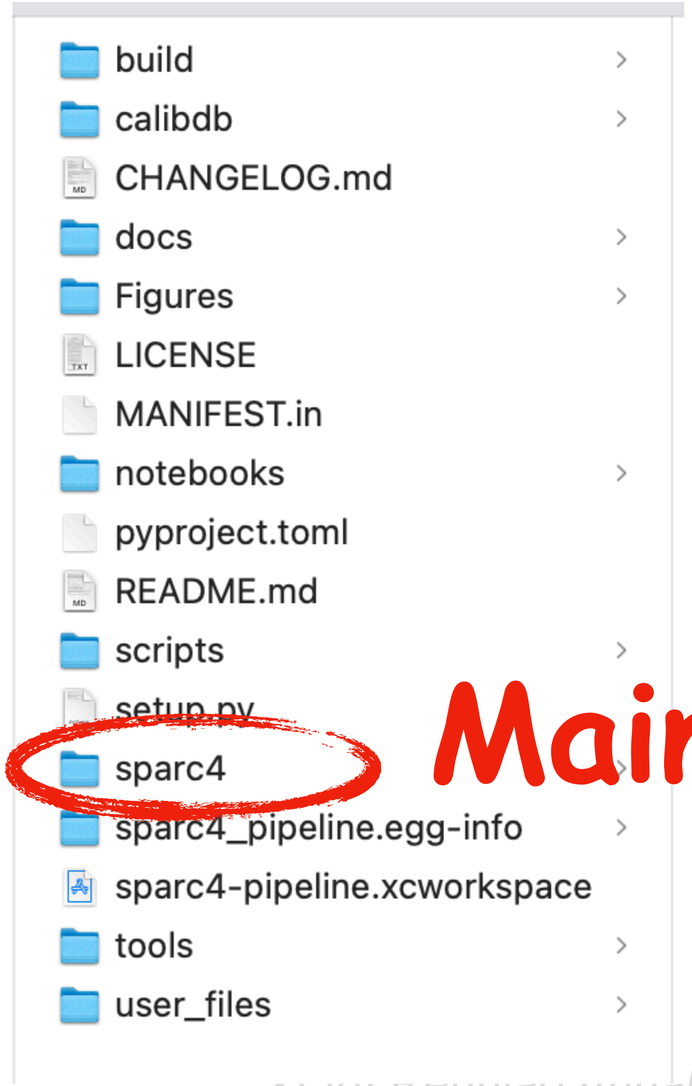
Pipeline concept



Customized use of the sparc4-pipeline

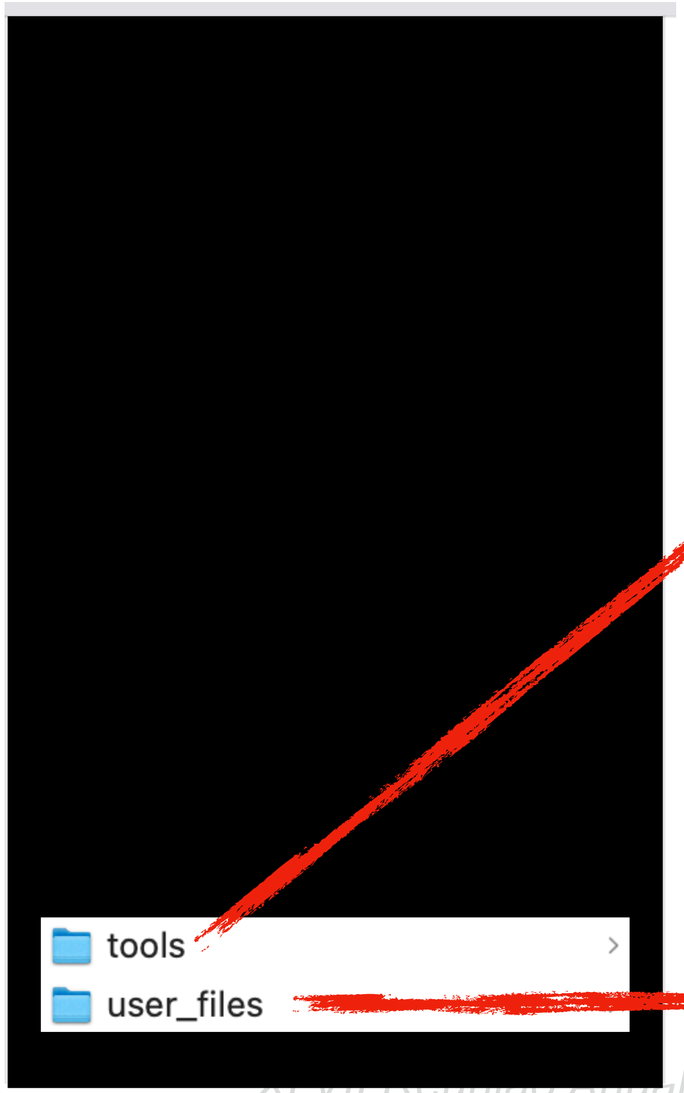
```
import sparc4.product_plots as s4plt
import sparc4.pipeline_lib as s4pipelib
import sparc4.utils as s4utils
import sparc4.db as s4db
import sparc4.products as s4p
```

Code structure



Main routines

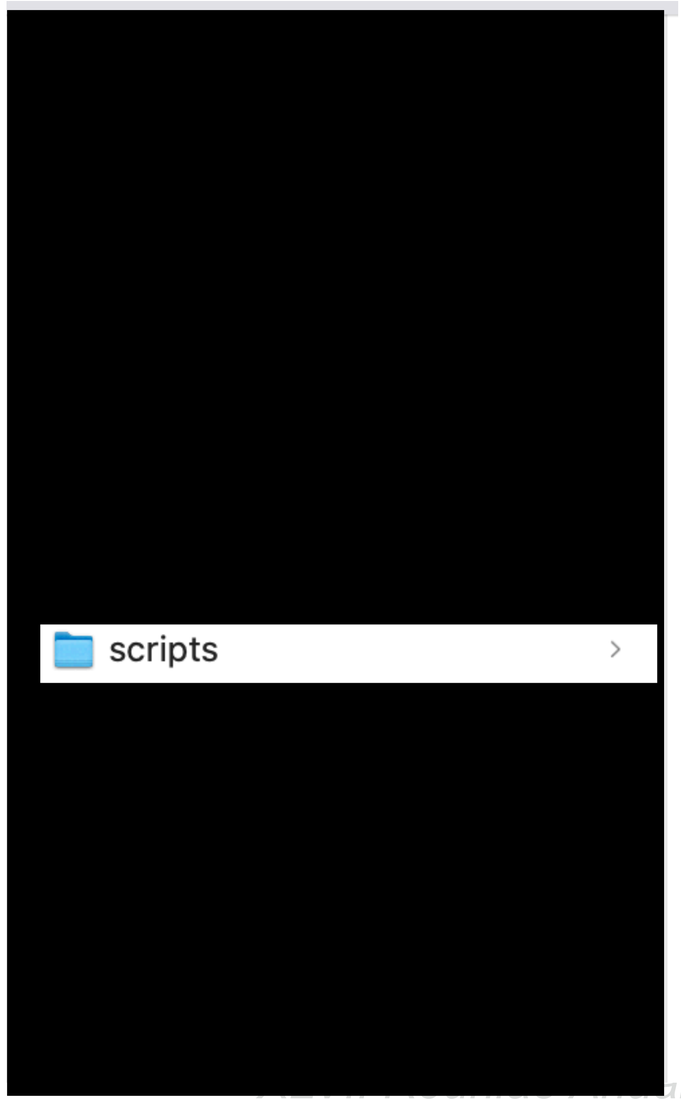
Code structure



- move_files_to_channels_inside_nights.py
- sparc4_distribute_data.py
- sparc4_fix_header.py
- sparc4_fix_wppos.py
- sparc4_focus.py
- sparc4_observed_objects.py
- sparc4_queue.py
- sparc4_quicklook.py

- my_params.yaml
- rolling_polar_params.yaml
- targets.csv

Code structure



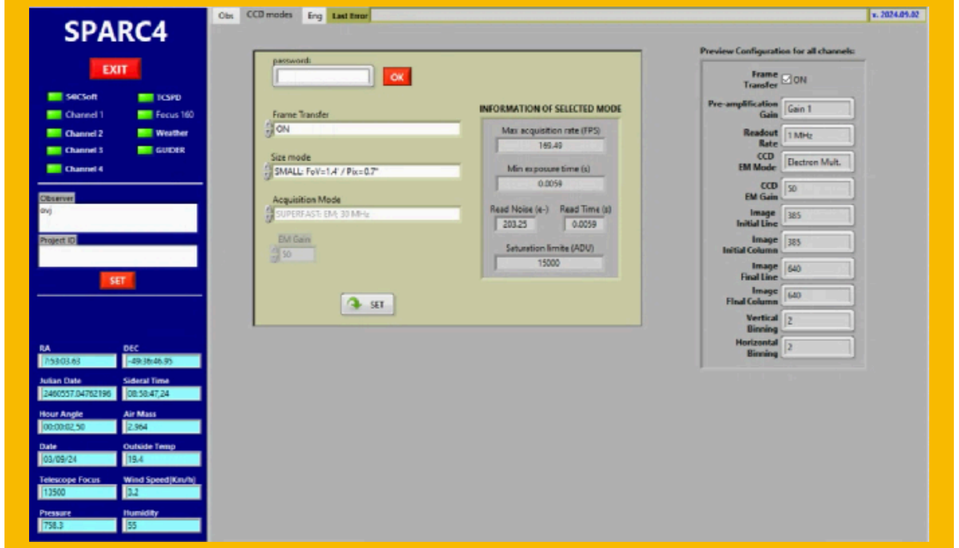
opd_mini_pipeline.py

sparc4_mini_pipeline.py

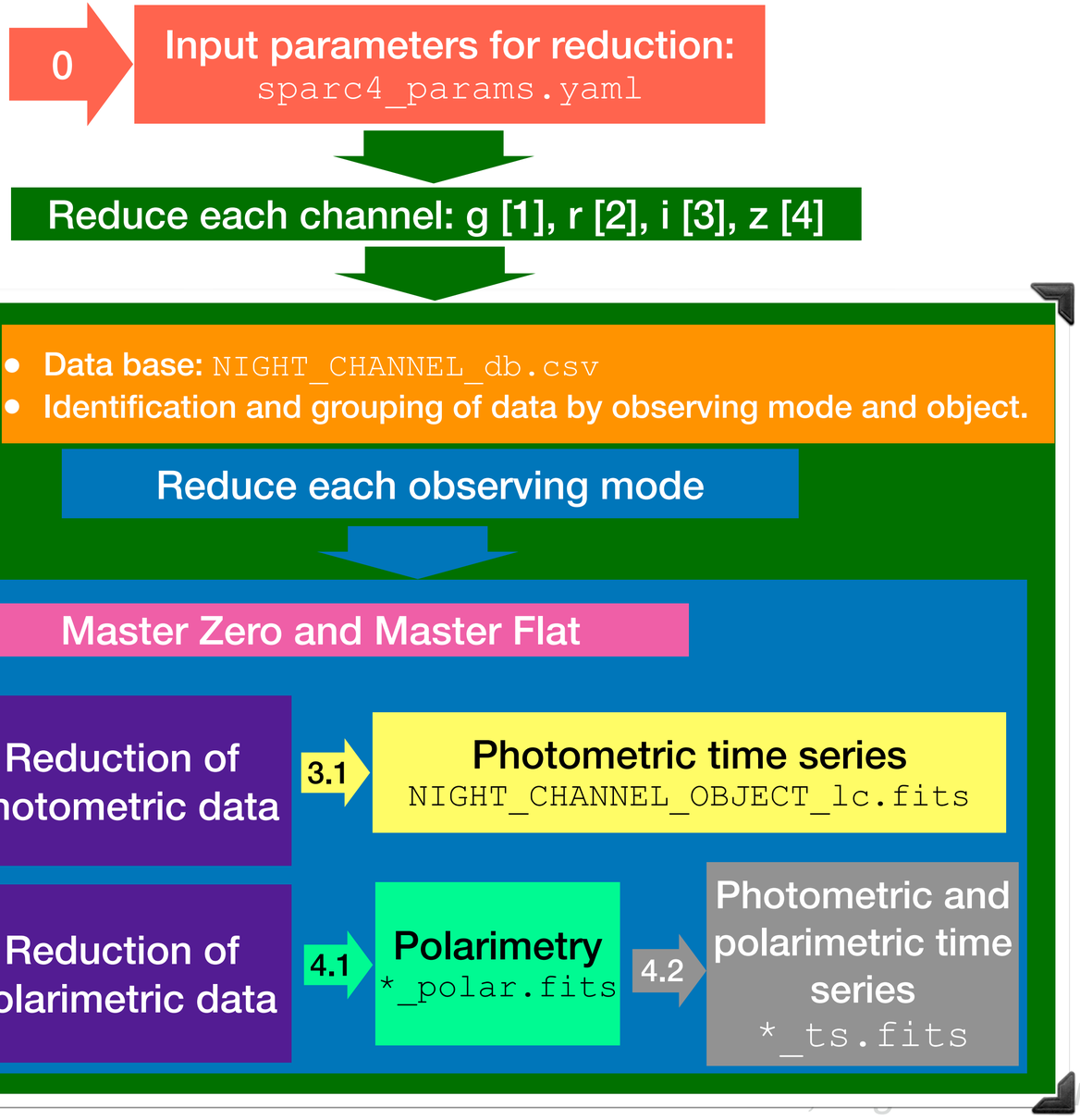
Observing modes

- **PHOT**
  - Imaging
  - Photometric time series (light curve)
- **POLAR**
  - $\lambda/2$ (L2) or $\lambda/4$ (L4) using up to 16 wave plate positions
  - Imaging
  - Polarimetric time series

- **Frame transfer ON/OFF**
- **ROI: 1024x1024, 512x512, 256x256**
- **Binning: 1x1, 2x2**
- **Conv: 0.1 or 1 MHz. EM:10, 20, 30 MHz**



Reduction overview



Basic reduction (steps 3 and 4 — all modes):

1. CCD gain correction;
2. Master Zero subtraction;
3. Master Flat correction;
4. Image registering;
5. Build stack: `NIGHT_CHANNEL_CCDMODE_OBJECT_INSTMODE_stack.fits`
6. Source detection - build master catalog;
7. Astrometry: new WCS;
8. Multi-aperture photometry;
9. Calculation of BJD and HJD;
10. Save data in a FITS file `*_proc.fits` (reduced image, catalogs, and new calibrations).

Installation

Install dependencies:

- Anaconda3 (recommended)
- AstroPoP 0.9.X
- astropy, astroquery, photutils, aafitrans, twirl, yaml, regions, uncertainties, time, signals, glob, functools
- astrometry.net (optional). Require index files: http://data.astrometry.net (for SPARC4 we use only **index-5200** a **index-5205**)

Obtaining the sparc4-pipeline:

- Download SPARC4 Pipeline package in the link sparc4-pipeline or
- On github: https://github.com/edermartioli/sparc4-pipeline. Use the following git command:

```
git clone https://github.com/edermartioli/sparc4-pipeline.git
```

- Download the minidata package with SPARC4 data to test the pipeline.

2. Preparatory steps

Directory structure

Raw data:

```
$ROOTDATADIR/sparc4acs1/$NIGHTDIR/
$ROOTDATADIR/sparc4acs2/$NIGHTDIR/
$ROOTDATADIR/sparc4acs3/$NIGHTDIR/
$ROOTDATADIR/sparc4acs4/$NIGHTDIR/
```

$ROOTDATADIR is the root directory where raw data are saved

$**NIGHTDIR** is the night directory of observations

Reduced data (nights inside channels) :

```
$ROOTREDUCEDIR/sparc4acs1/$NIGHTDIR/
$ROOTREDUCEDIR/sparc4acs2/$NIGHTDIR/
$ROOTREDUCEDIR/sparc4acs3/$NIGHTDIR/
$ROOTREDUCEDIR/sparc4acs4/$NIGHTDIR/
```

Reduced data (channels inside nights) :

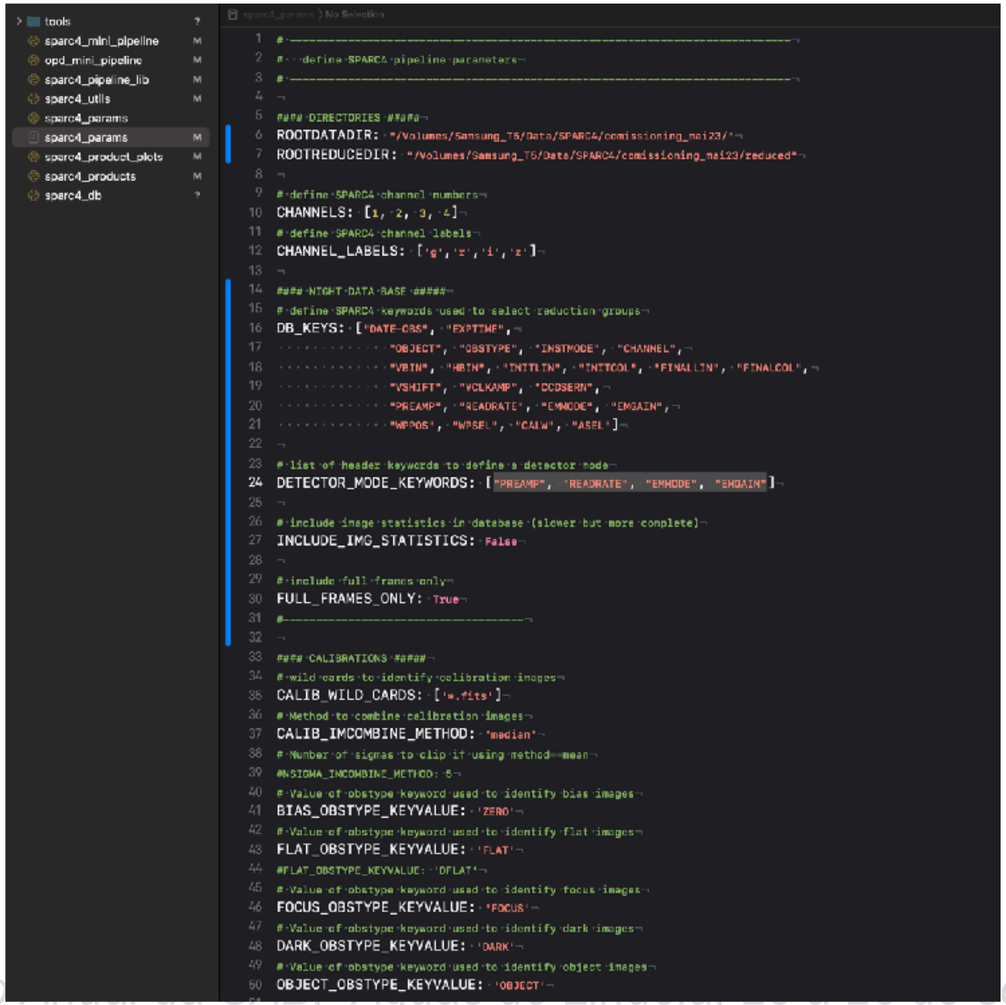
```
$ROOTREDUCEDIR/$NIGHTDIR/sparc4acs1/
$ROOTREDUCEDIR/$NIGHTDIR/sparc4acs2/
$ROOTREDUCEDIR/$NIGHTDIR/sparc4acs3/
$ROOTREDUCEDIR/$NIGHTDIR/sparc4acs4/
```

$ROOTREDUCEDIR is the root directory where reduced data are saved

Pipeline configuration

Create a new parameter file and edit it. Make a copy of the file:

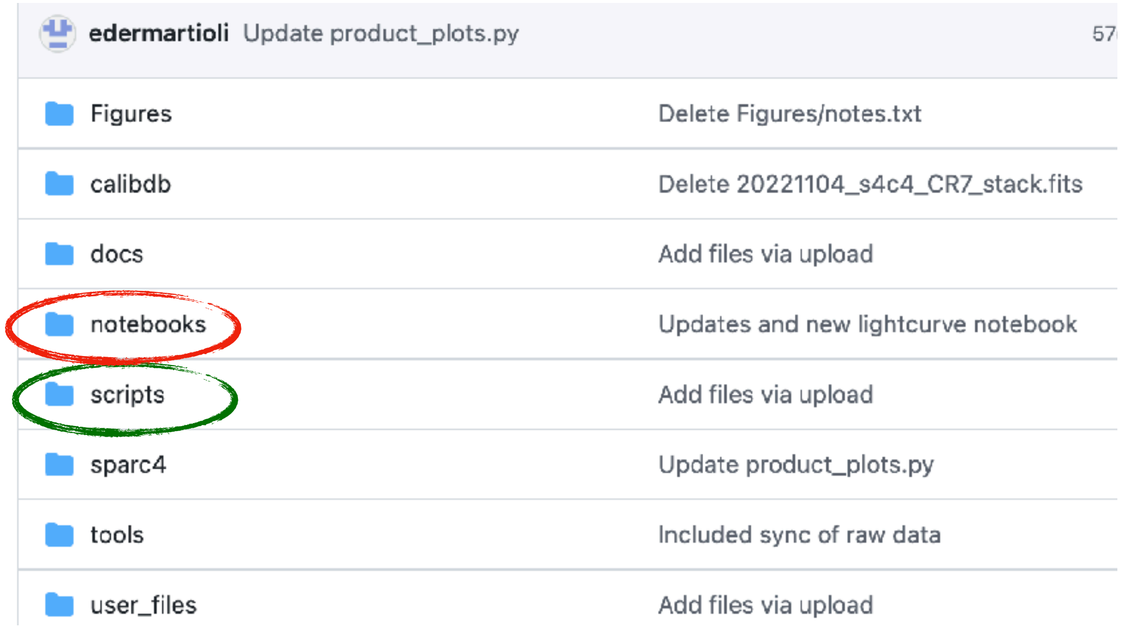
`sparc4-pipeline/user_files/my_params.yaml`



3. Reduction of minidata

I suggest two ways to reduce your data

- Run pipeline script in the command line
- Run pipeline functions in a Jupyter notebook or make your own scripts.



Execution in the terminal

```
python sparc4_mini_pipeline.py …
```

```
--datadir='/Users/eder/Data/SPARC4/minidata/'     (Data root directory)

--reducedir='/Users/eder/Data/SPARC4/minidata/reduced/'   (Reduced data root directory)

--channels='1,3'   (Seletected SPARC4 channels)

--nightdir='20230604'   (Name of night directory common to all channels)

--object='Hilt 652'   (Object ID – same as in header key OBJECT – to restrict reduction)

--params='/Users/eder/sparc4-pipeline/user_files/my_params.yaml' (Input yaml parameter file)

--target_list='/Users/eder/sparc4-pipeline/user_files/targets.csv' (Input target list)

-f    (Force reduction)

-p    (Activate plotting)
```

Execution in the terminal

```
python sparc4_mini_pipeline.py --nightdir=20230604
```

```
python -W ignore /Users/eder/sparc4-pipeline/
scripts/sparc4_mini_pipeline.py --params=/Users/
eder/Desktop/my_params.yaml --nightdir=20230604
```

How long it takes to reduce the minidata?

On a MacBook Pro 2023 M2 Max 32Gb :

- 20230604 : 10m44s
- 20230605 : 2m09s
- 20230606 : 9m27s
- 20240618 : 3m50s

Break

SPARC4 reduction products

- **NIGHT+CHANNEL DATABASE (`db.csv`)** — night database;
- **MASTER ZERO (`MasterZero.fits`)** —combined image of all zero images;
- **MASTER FLAT (`MasterDomeFlat.fits`)** — combined image of all flat images;

- **PROCESSED FRAME (`proc.fits`)** —calibrated scientific image;
- **STACK FRAME (`stack.fits`)** — combined image of a number of scientific processed images;
- **PHOTOMETRIC LIGHT CURVE (`lc.fits`)** — time series file containing all photometric quantities;
- **POLARIMETRY (`polar.fits`)** — polarimetry results for all sources obtained from a sequence of many λ/2 or λ/4 wave plate positions;
- **POLARIMETRIC TIME SERIES (`ts.fits`)** — time series file containing all photometric and polarimetric quantities.