USING PypeIt TO REDUCE OSIRIS+/MOS DATA

Marta Lorenzo
Centro de Astrobiología (CAB), CSIC-INTA,
Torrejón de Ardoz (Madrid), Spain
mlorenzo@cab.inta-csic.es

RIA

MINISTERIO
DE CIENCIA
E INNOVACIÓN

AGENCIA
ESTATAL DE
INVESTIGACIÓN

CENTRO DE ASTROBIOLOGÍA
CSIC - INTA

EXCELENCIA
MARÍA
DE MAEZTU

UAM
Universidad Autónoma
de Madrid

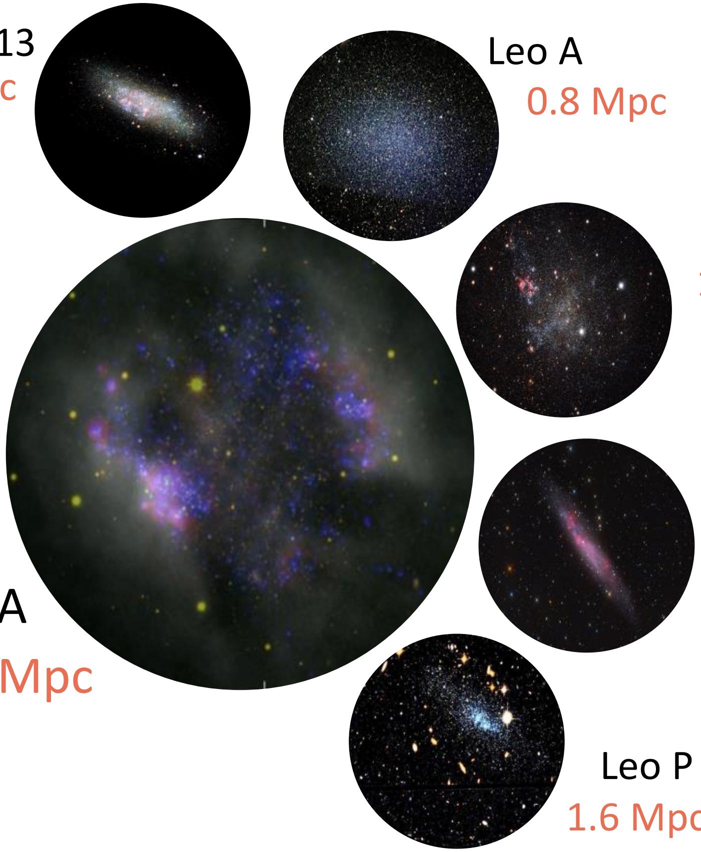
With the support of grants: PID2019-105552RB-C41, MDM-2017-0737 Unidad de Excelencia "María de Maeztu"-Centro de Astrobiología (CSIC-INTA), funded by MCIN/AEI/10.13039/501100011033, PRE2019-087988 under project MDM-2017-0737-19-3, and “ESF Investing in your future“.

A BIT OF CONTEXT...

Study massive stars in galaxies...

relatively close

extremely metal-poor



IC 1613
0.7 Mpc

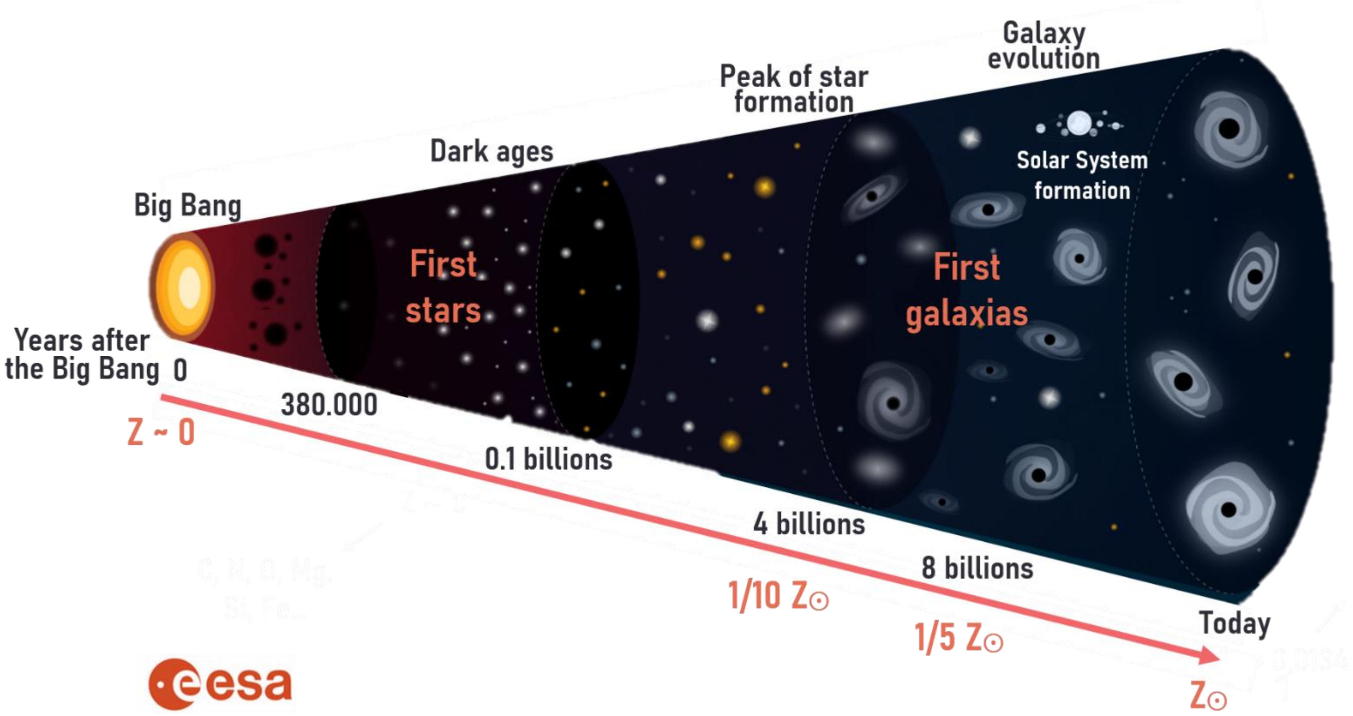
Leo A
0.8 Mpc

WLM
1.0 Mpc

Sextans A
1.3 Mpc

NGC 3109
1.3 Mpc

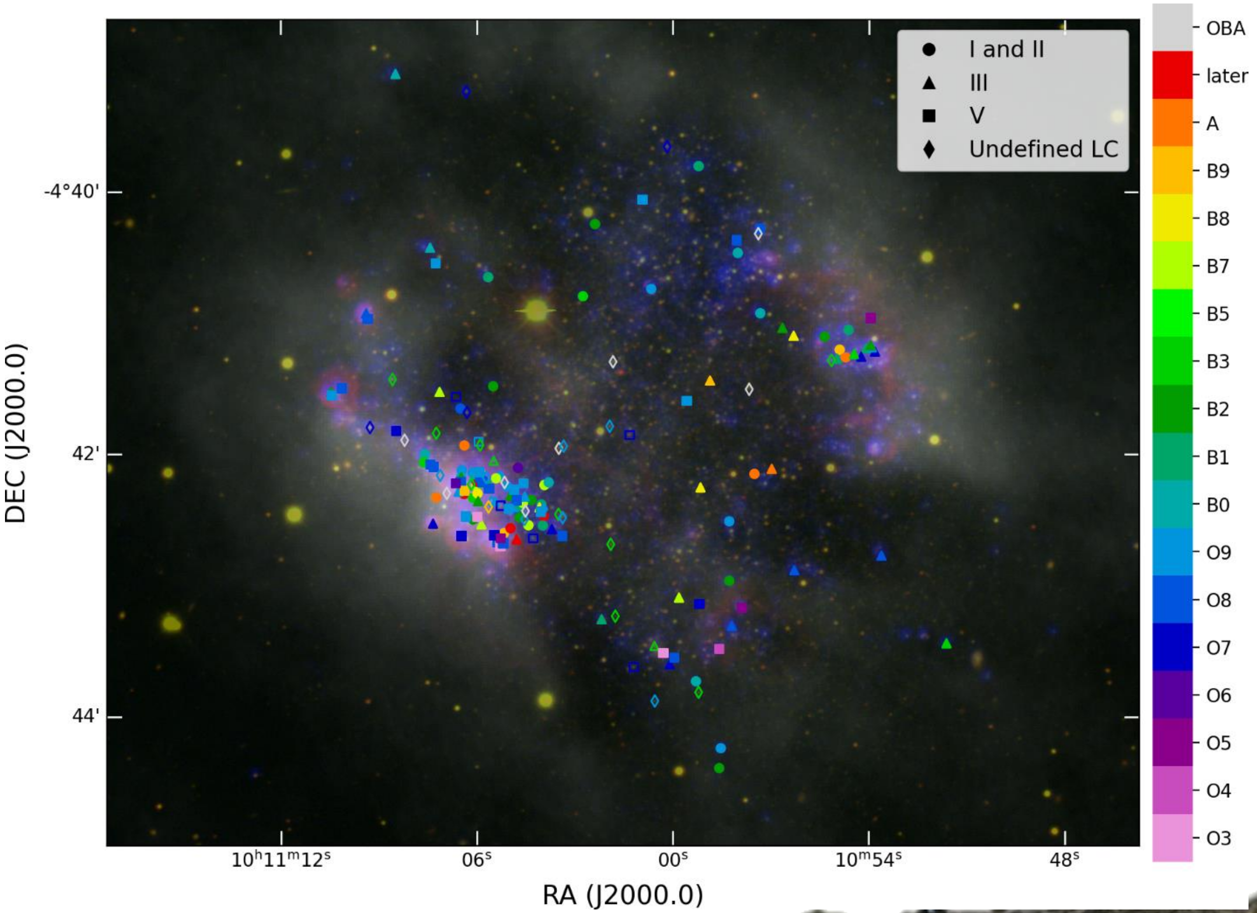
Leo P
1.6 Mpc



A BIT OF CONTEXT...

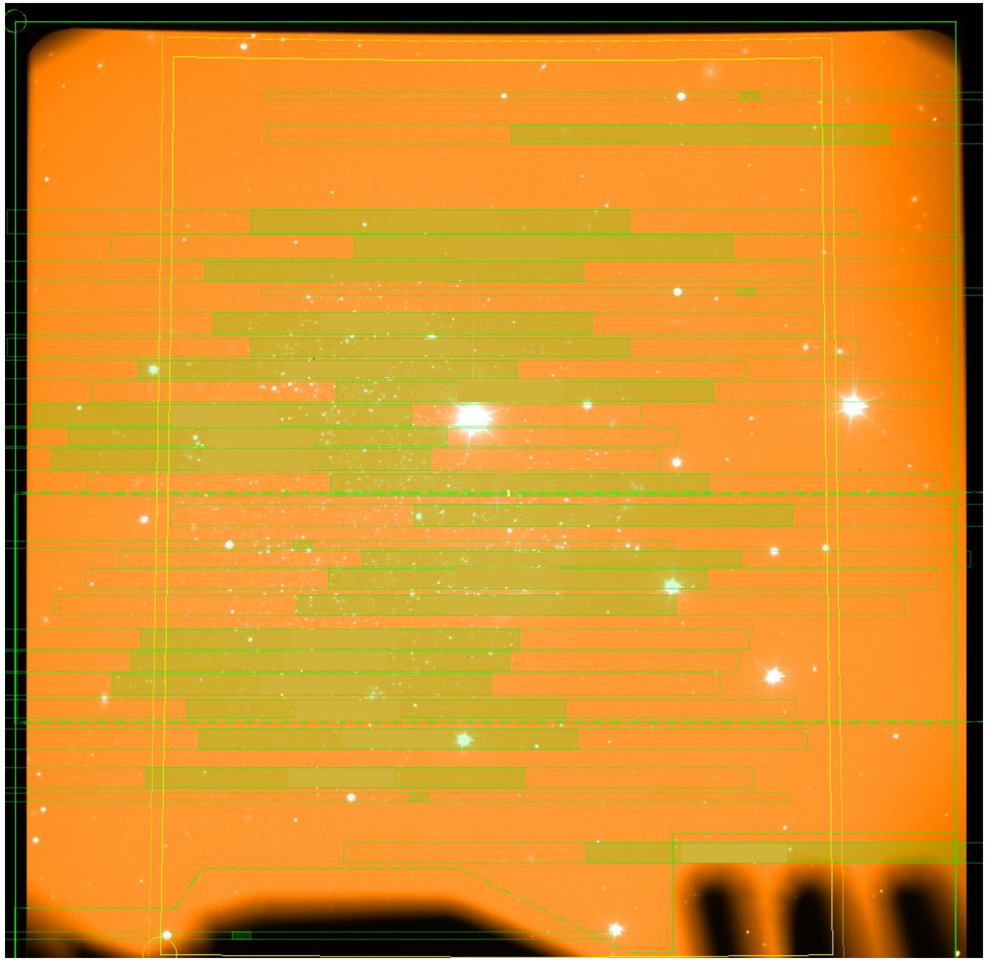
THE **LARGEST** SPECTROSCOPIC CATALOGUE OF EXTREMELY METAL-POOR MASSIVE STARS

Lorenzo et al. (2022), MNRAS, 516, 3



A BIT OF CONTEXT...

19h with OSIRIS+/MOS



MAIN GOAL

Constraining the stellar parameters and evolution of our targets

↓

Spectroscopy with high resolution and high S/N of faint O stars

SECONDARY GOAL

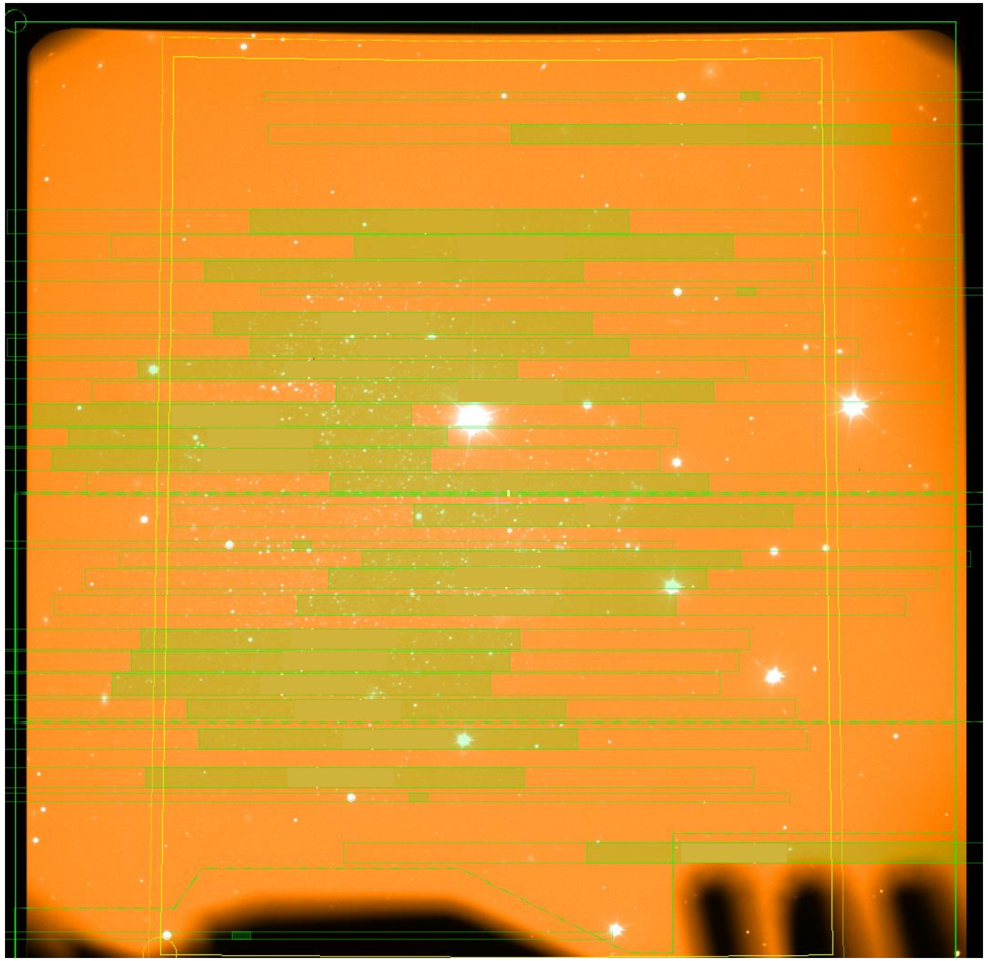
Estimating the stellar abundances of Sextans A

↓

Spectroscopy with high resolution and high S/N of BA supergiants

A BIT OF CONTEXT…

~~19h~~ 2 h with OSIRIS+/MOS



~~MAIN GOAL~~

~~Constraining the stellar parameters and evolution of our targets~~

↓

~~Spectroscopy with high resolution and high S/N of faint O stars~~

SECONDARY GOAL

Estimating the stellar abundances of Sextans A

↓

Spectroscopy with high resolution and high S/N of BA supergiants

DEFAULT REDUCTION WITH PypeIt

1. Create directories:
   - RAWDIR/
   - RDXDIR/

2. Setup PypeIt

3. Edit your PypeIt file

   default

4. Run PypeIt

5. Examine the results:
   - Calibrations
   - Reduced spectra

DEFAULT REDUCTION WITH PypeIt

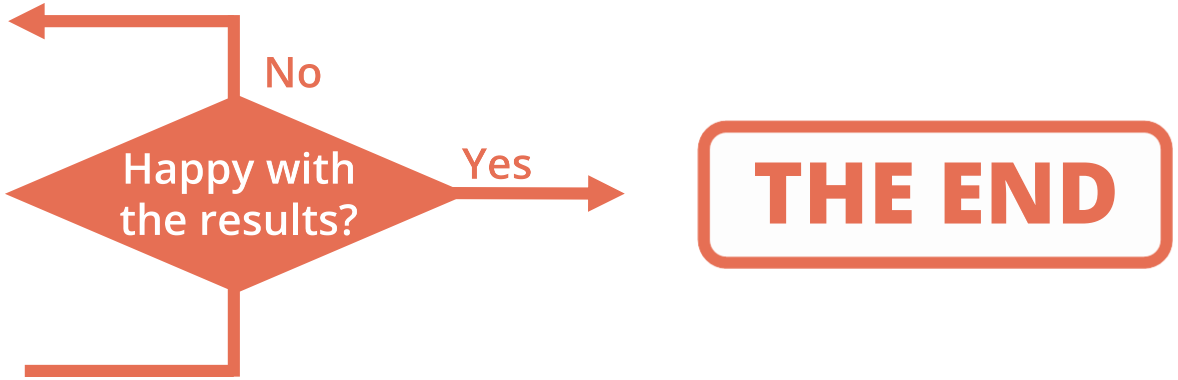
1. Create directories:
   - RAWDIR/
   - RDXDIR/

2. Setup PypeIt

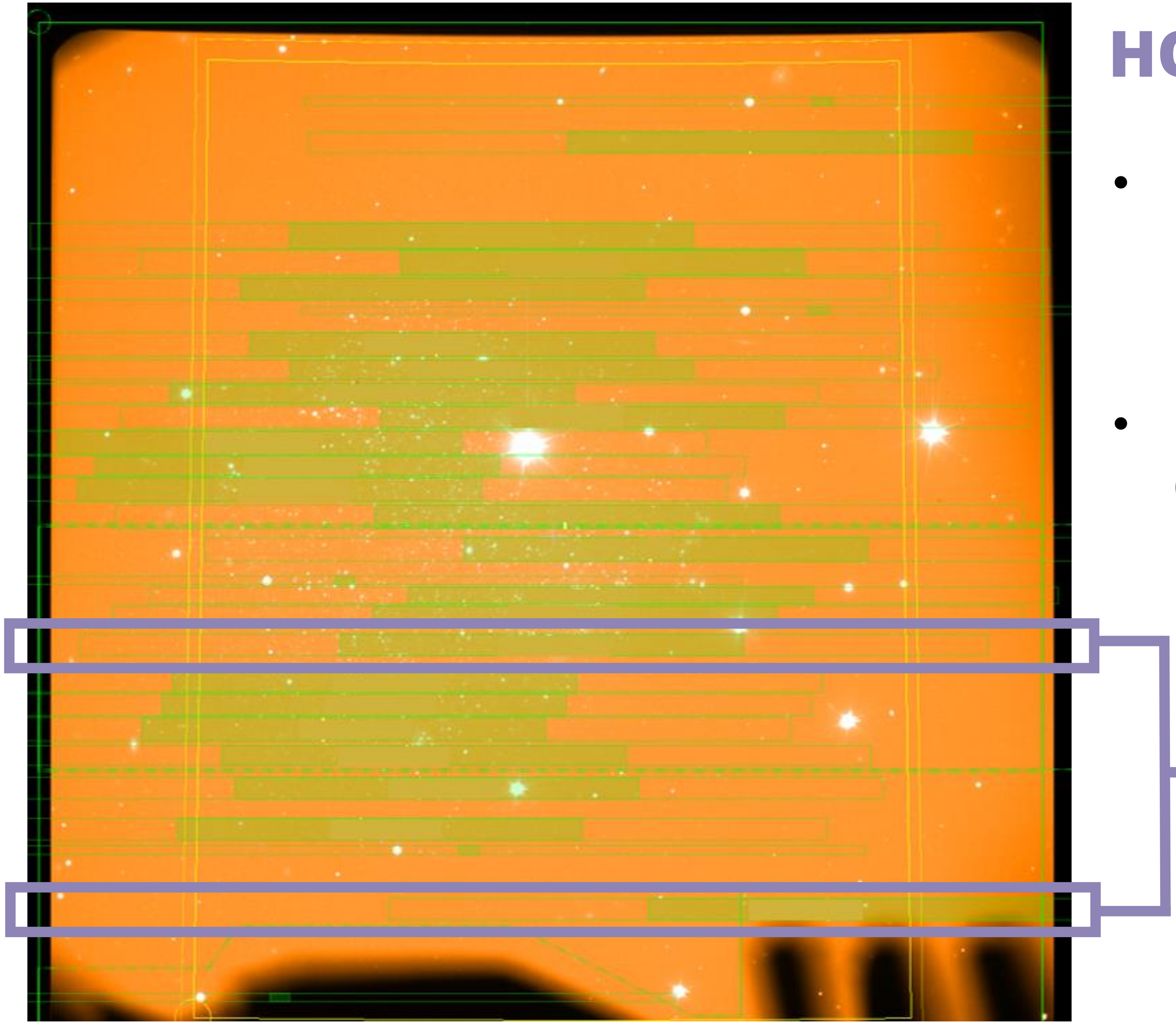
3. Edit your PypeIt file

4. Run PypeIt

5. Examine the results:
   - Calibrations
   - Reduced spectra



DETECTION OF SLITS:



HOW TO DETECT THIS PROBLEM

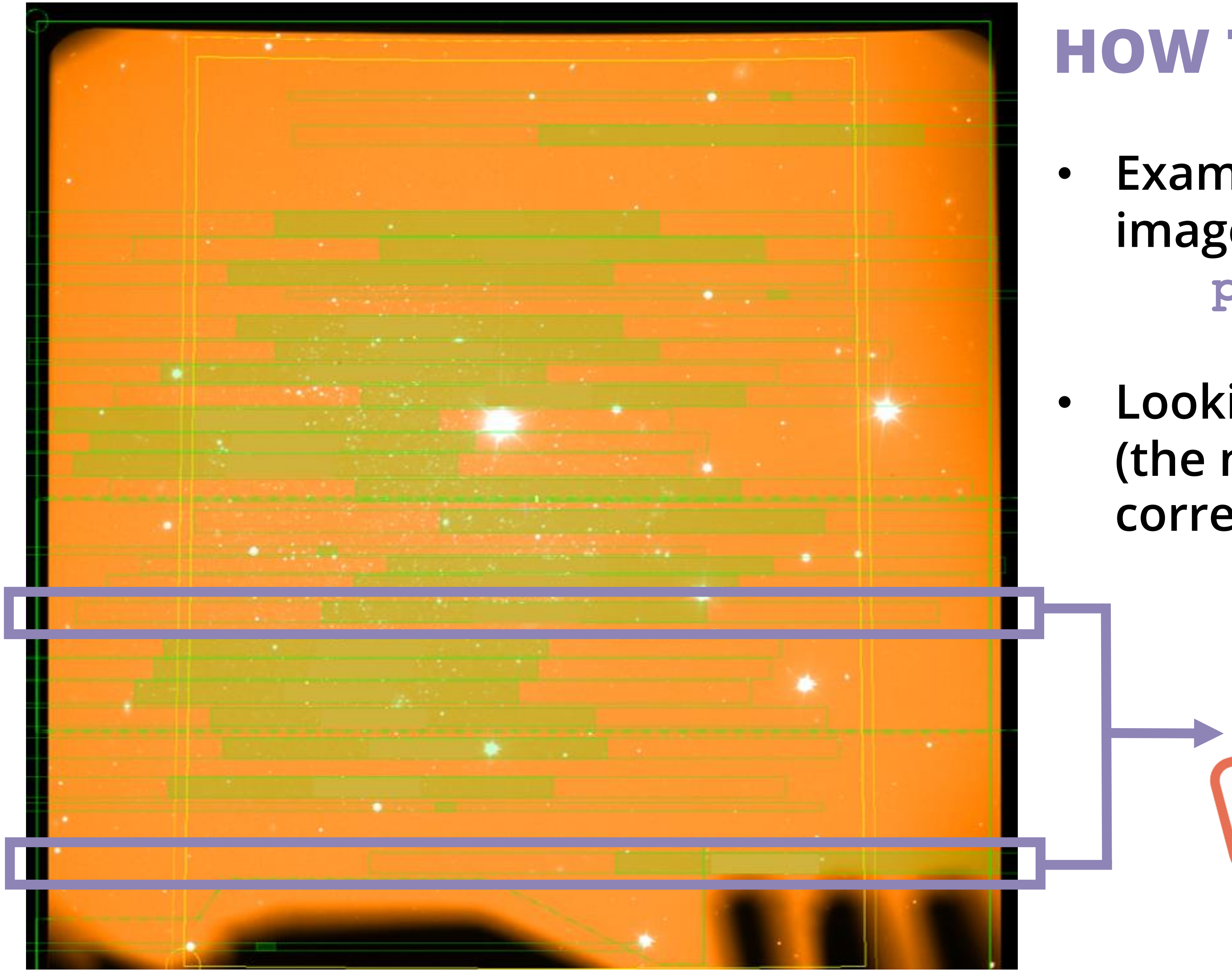
- Examining the edges and slit 2D-images:

  ```
  pypeyt_chk_edges -h
  ```

- Looking at the list of extracted sources (the name of the files contains the corresponding coordinates in the mask)

NOT DETECTED

DETECTION OF SLITS:



HOW TO DETECT THIS PROBLEM

- Examining the edges and slit 2D-images:

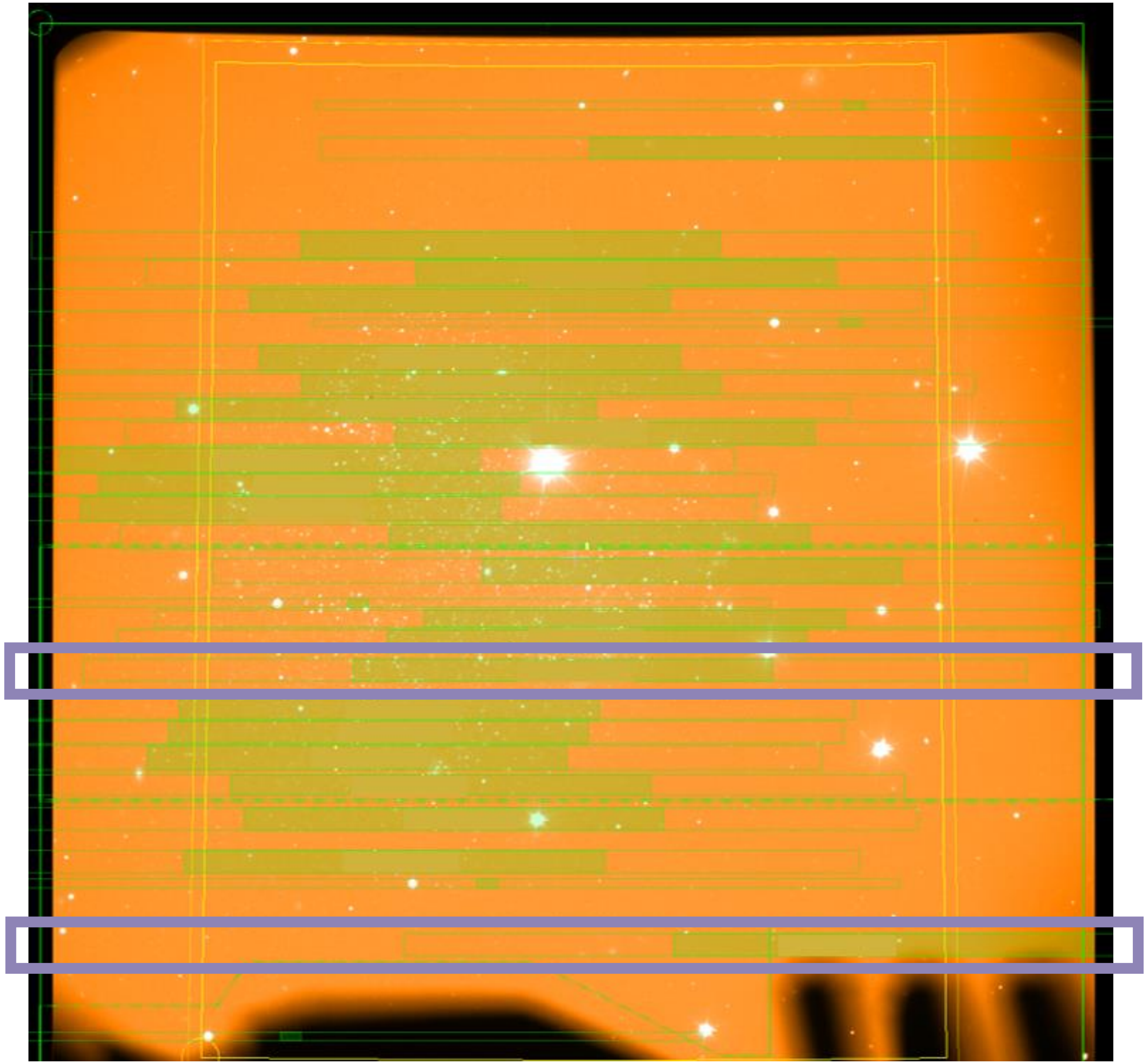
  ```
  pypeyt_chk_edges -h
  ```

- Looking at the list of extracted sources (the name of the files contains the corresponding coordinates in the mask)

NOT DETECTED

NOT RESOLVED

DETECTION OF SLITS:



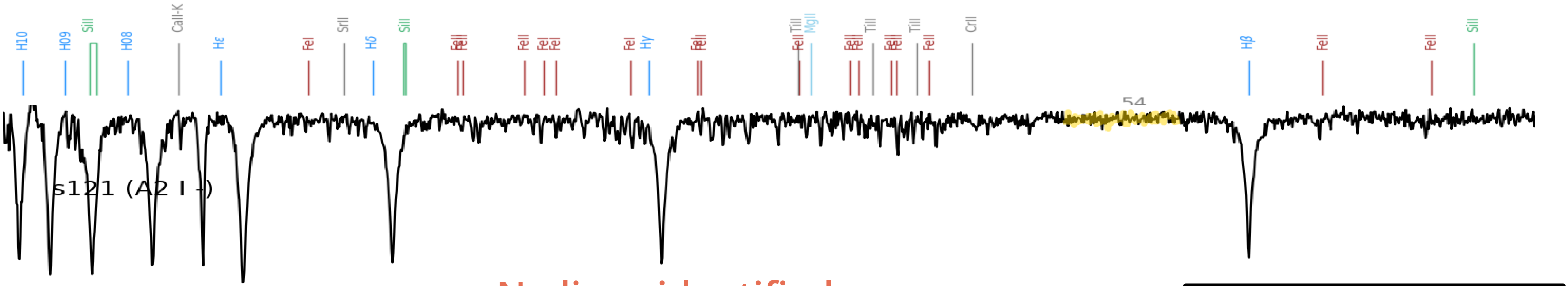
POSSIBLE SOLUTIONS

- Modify the `edge_thresh` or `minimum_slit_length` keywords in the Pypeit file.

- Manually add the missed slits:

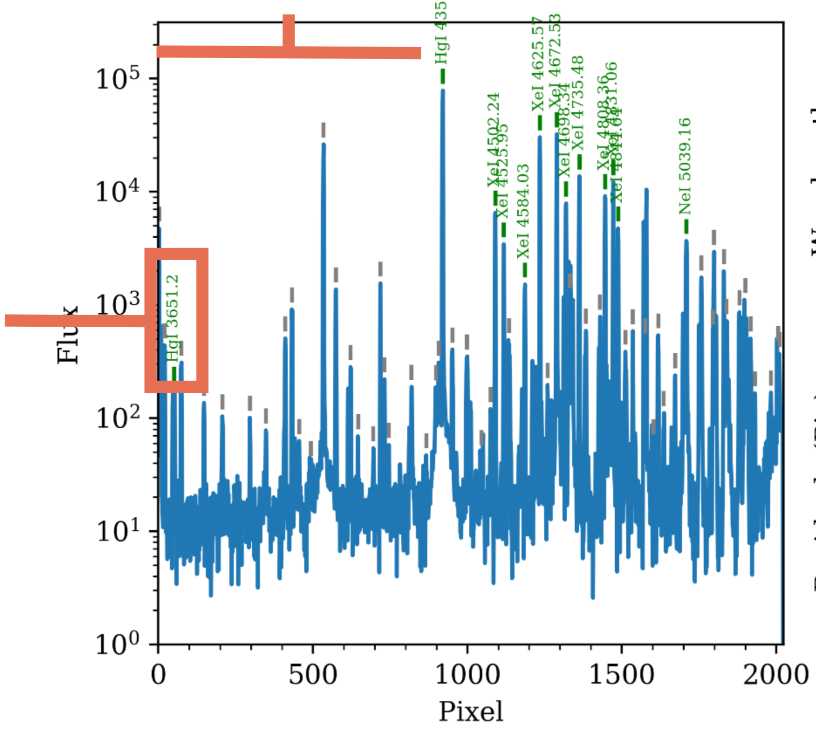
```
[calibrations]
    [[slitedges]]
        add_slits = (coords.)
```

WAVELENGTH CALIBRATION:

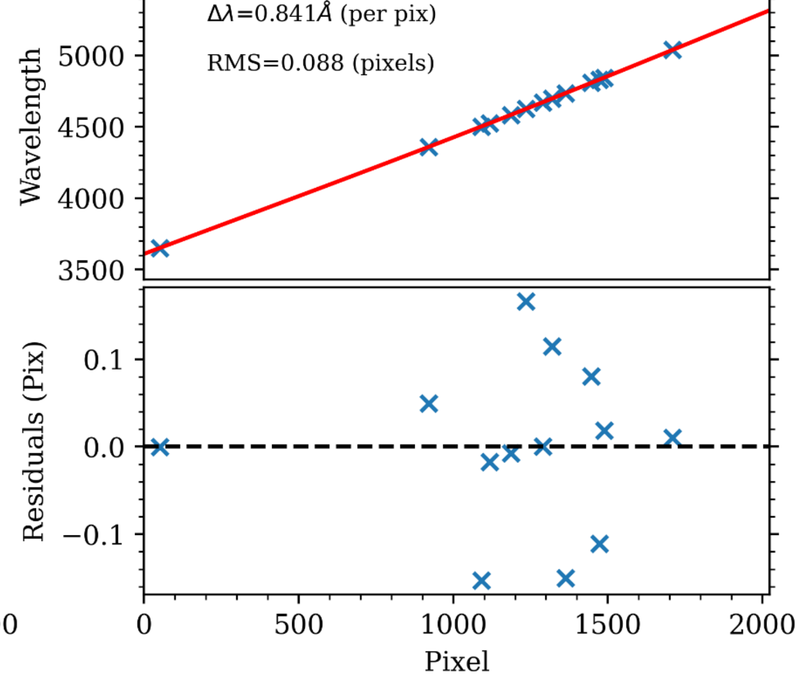


No lines identified

Wrong identification



In folder: QA/Arc_*



WAVELENGTH CALIBRATION:

SOLUTION

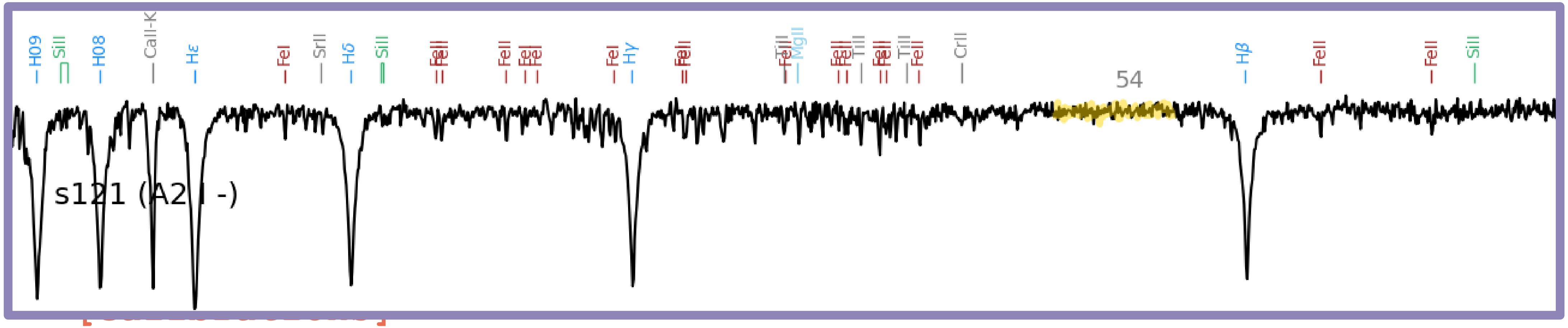
1. Use your own line lists:
    1. Download them
    2. Save them in the RDXDIR/
    3. Install line lists:
       ```
       pypeit_install_linelist ArI_lines.dat
       ```
    4. Introduce lines in .pypeit
       ```
       [calibrations]
         [[wavelengths]]
           lamps = ArI, CdI, HgI,HgCd_MMT
       ```
2. Identify lines manually:
    ```
    pypeit_identify Arc.fits Slits.fits.gz --slit #
    ```
3. Save results and introduce them in .pypeit

SPECIAL THANKS TO RYAN COOKE

WAVELENGTH CALIBRATION:

SOLUTION

1. Use your own line lists:



```
  [[wavelengths]]
    lamps = ArI, CdI, HgI,HgCd_MMT
```

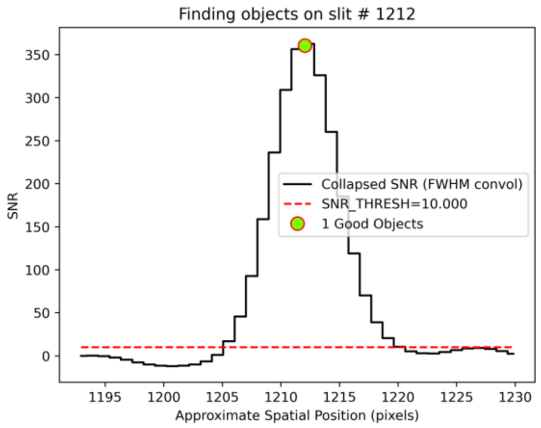
SPECIAL THANKS TO RYAN COOKE

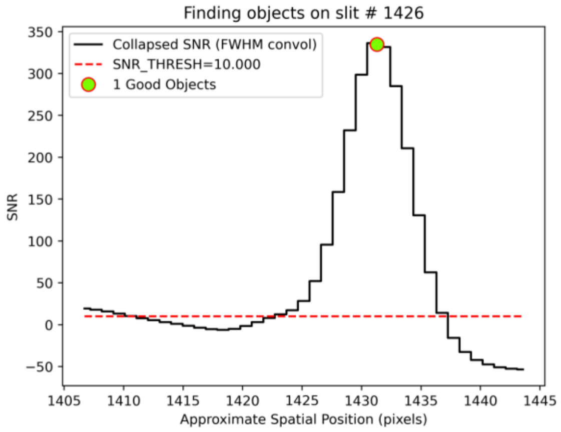
2. Identify lines manually:

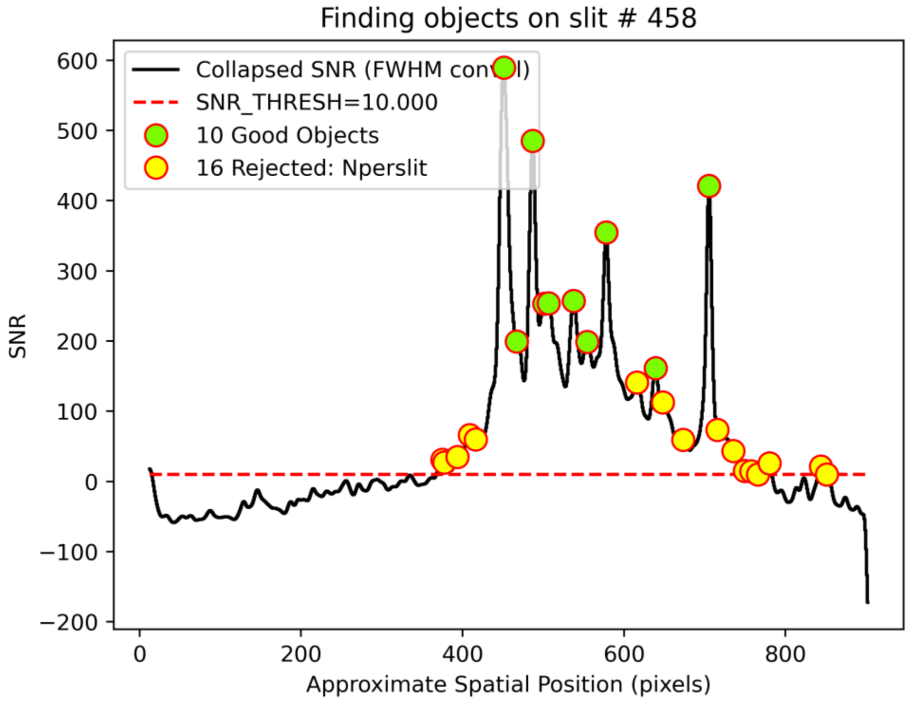
```
pypeit_identify Arc.fits Slits.fits.gz --slit #
```

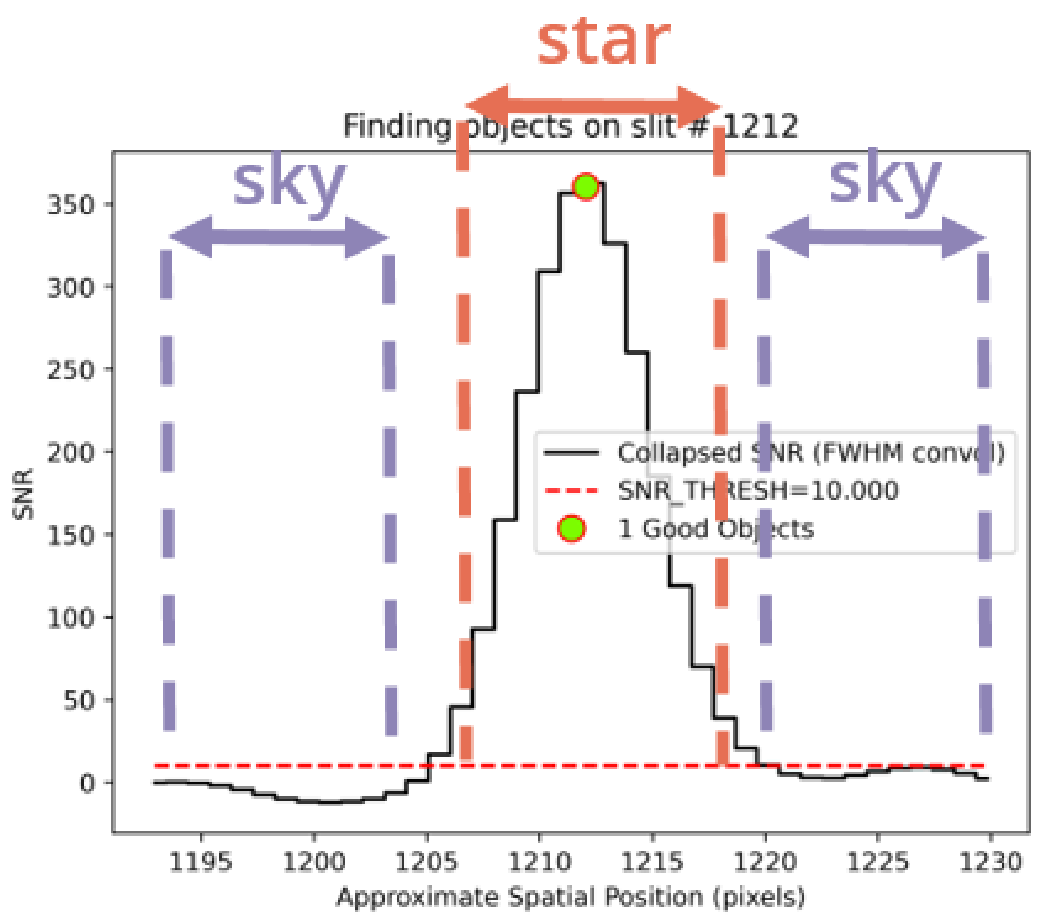
3. Save results and introduce them in .pypeit

SUPERVISING THE EXTRACTION:









NEED OF THIS INFO

FURTHER PROCESSING WITH PypeIt

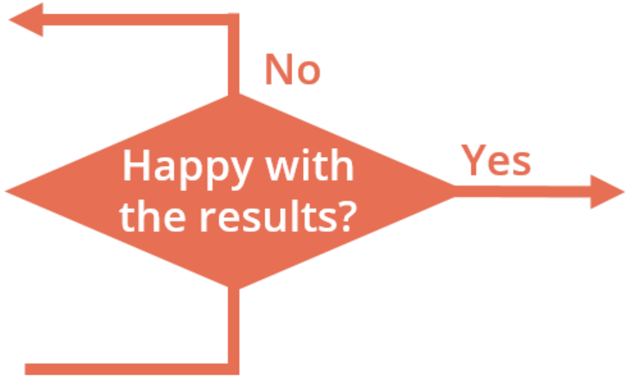
1. Create directories:
   - RAWDIR/
   - RDXDIR/

2. Setup PypeIt

3. Edit your PypeIt file

4. Run PypeIt

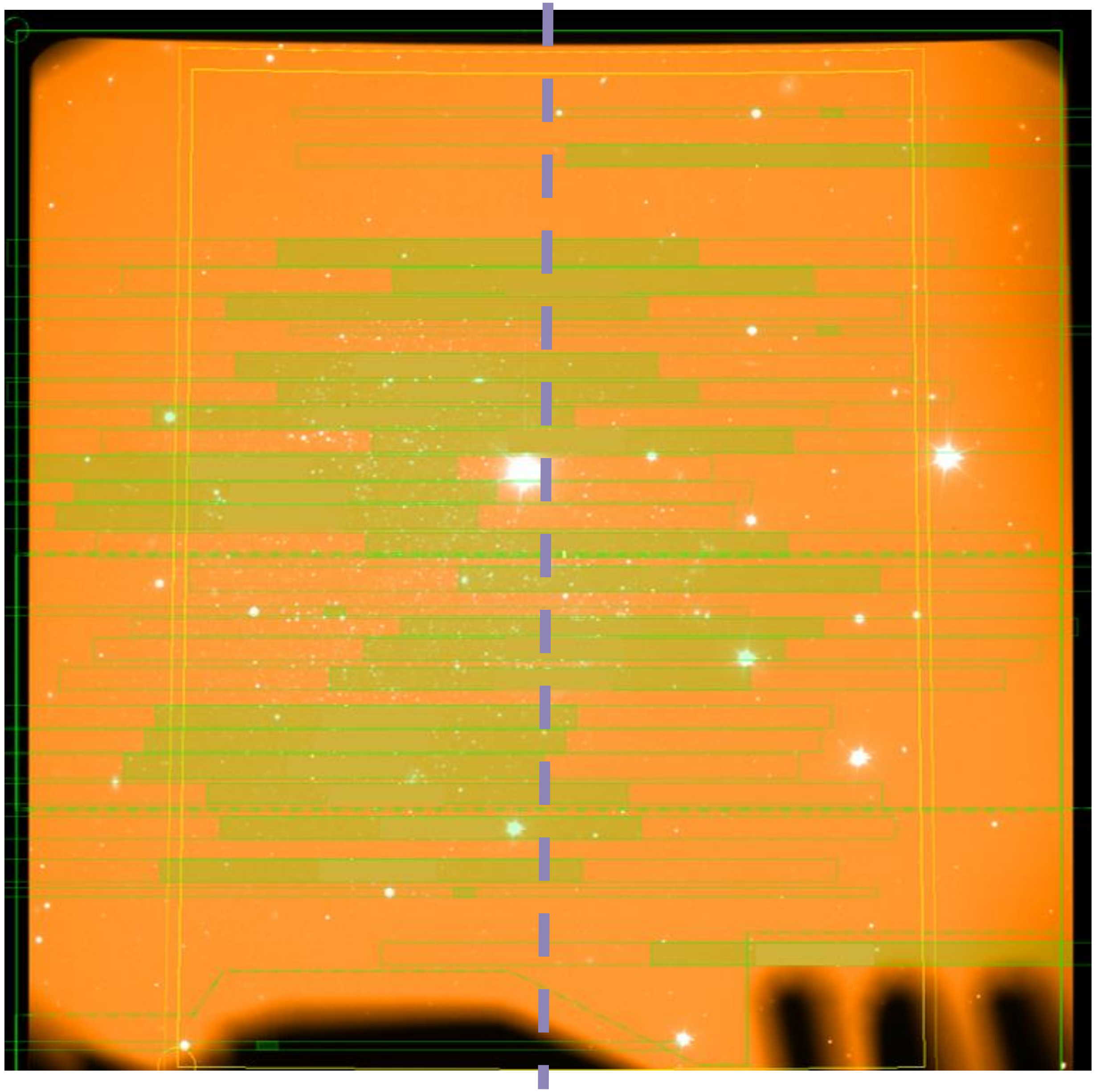
5. Examine the results:
   - Calibrations
   - Reduced spectra



*. Flux calibration

*. Coaddition of 1D spectra

FURTHER PROCESSING WITH PypeIt: FLUX CALIBRATION



IF YOUR SLITS ARE APPROX. CENTERED IN THE MASK

1. Creating a pypeit sensitivity function:

```
pypeit_sensfunc standard.fits
-o sens.fits
```

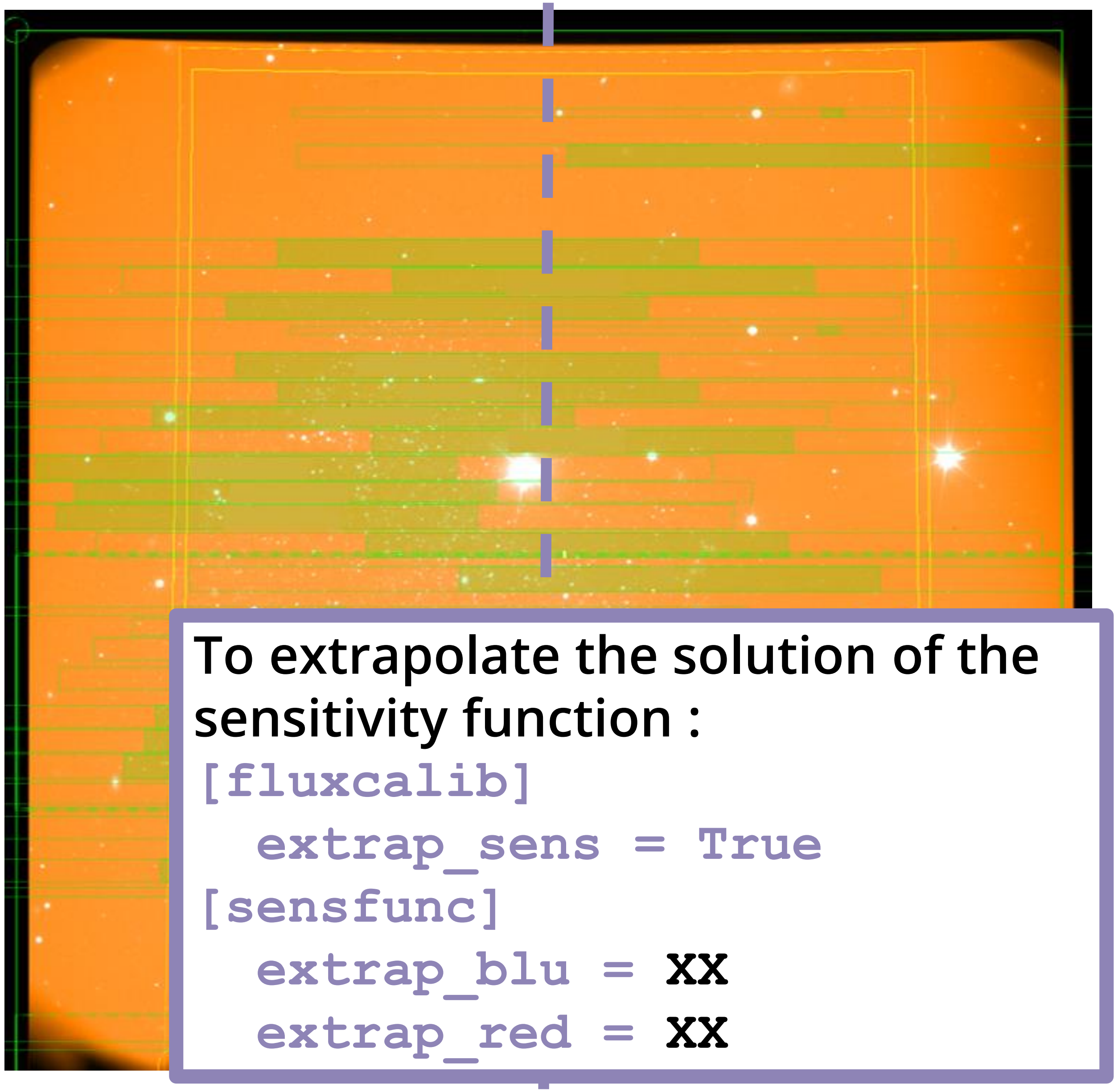
2. Creating a flux file

```
flux read
     filename | sensfile
flux end
```

3. Calibrating in flux:

```
pypeit_flux_calib fluxfile.txt
```

FURTHER PROCESSING WITH PypeIt: FLUX CALIBRATION



To extrapolate the solution of the sensitivity function :

```
[fluxcalib]
  extrap_sens = True
[sensfunc]
  extrap_blu = XX
  extrap_red = XX
```

IF YOUR SLITS ARE APPROX. CENTERED IN THE MASK

1. Creating a pypeit sensitivity function:

```
pypeit_sensfunc standard.fits
-o sens.fits
```

2. Creating a flux file

```
flux read
     filename | sensfile
flux end
```

3. Calibrating in flux:

```
pypeit_flux_calib fluxfile.txt
```

FURTHER PROCESSING WITH PypeIt: COADDITION

1. Specify whether the spectra was or not calibrated:

```
[coadd1d]
     flux_value = True/False
```

2. Creating a coadd file

```
pypeit_collate_1d
```

```
coadd1d read
      filename | obj_id
coadd1d end
```

3. Coadd the spectra:

```
pypeit_coadd_1dspec coadd_file.txt
```

Good results with the default method:

```
[coadd1d]
   scale_method = auto
```

CONCLUSIONS

PROS vs. CONS

Fast to learn and apply

Black box in the sky subtraction and extraction steps

(cannot supervise or correct manually)

Fast and efficient response of the Pypeit team

SPECIAL THANKS TO RYAN COOKE

PYPEIT WILL BE MY GO-TO PIPELINE FOR REDUCING OSIRIS+ SPECTRA