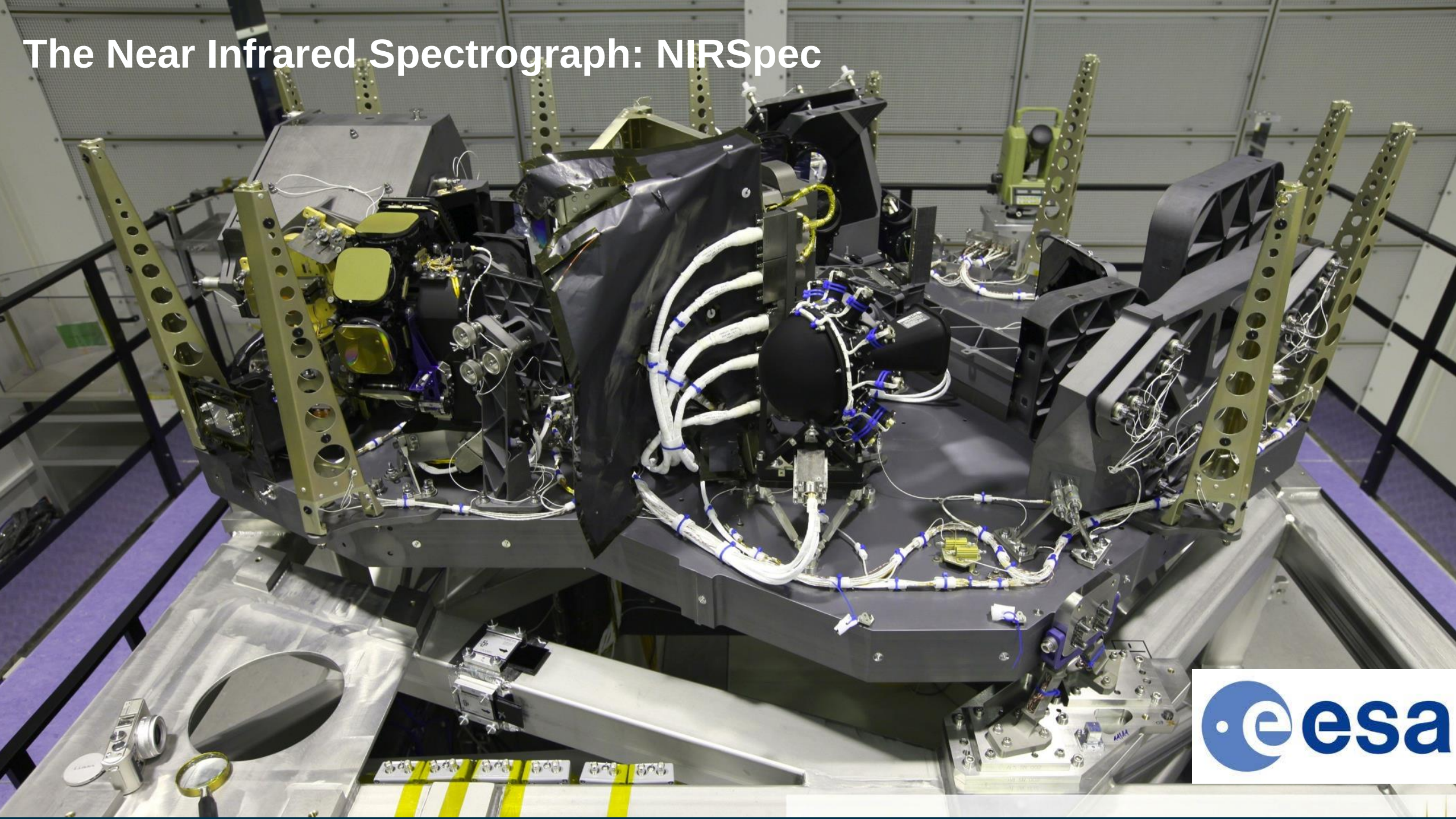
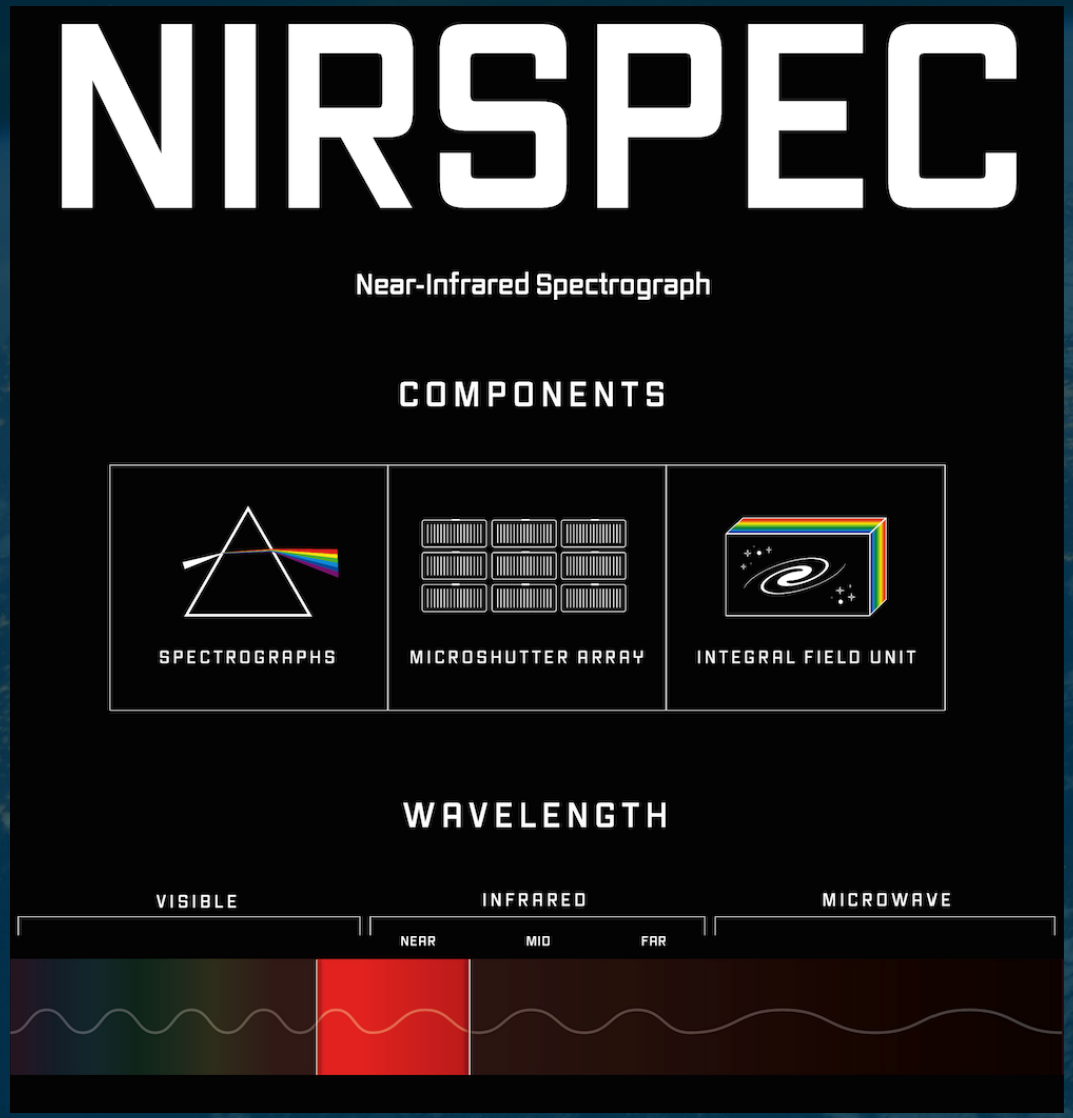
Fast turn-around reduction of NIRSpec MOS data in Datalabs

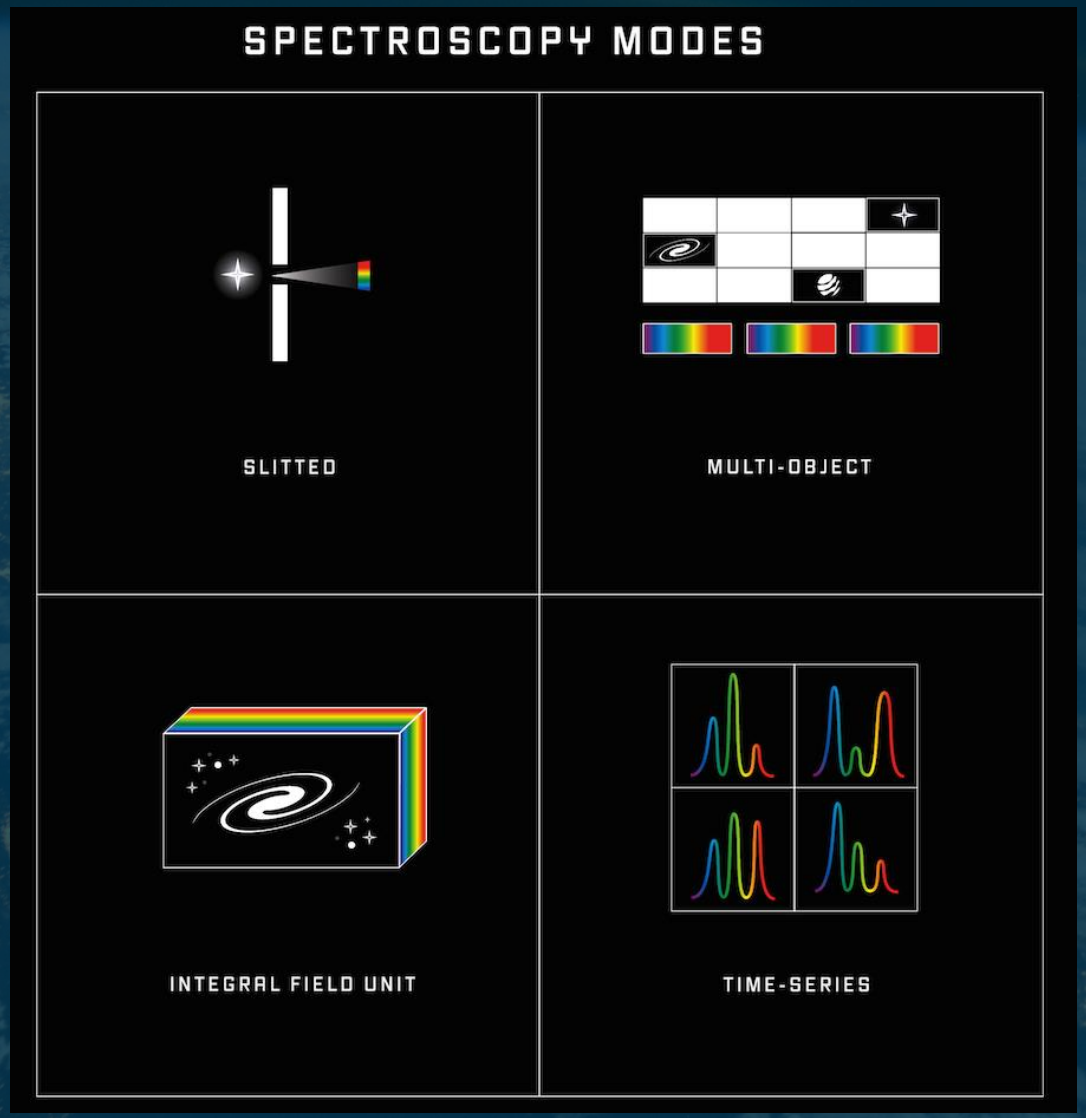
Giovanna Giardino

The Near Infrared Spectrograph: NIRSpec



NIRSpec supports 4 different observing modes



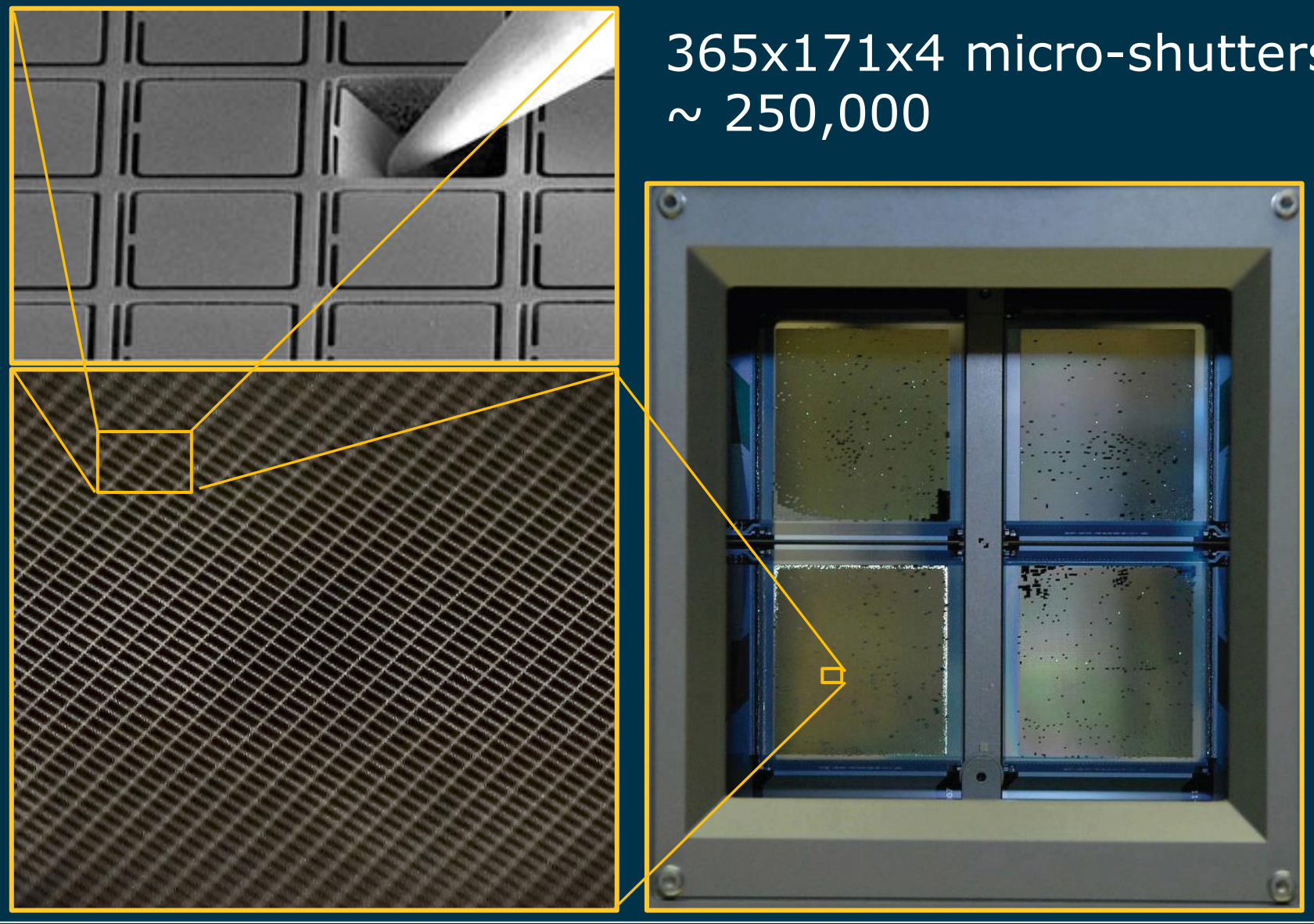


Micro Shutters Assembly of NIRSpec

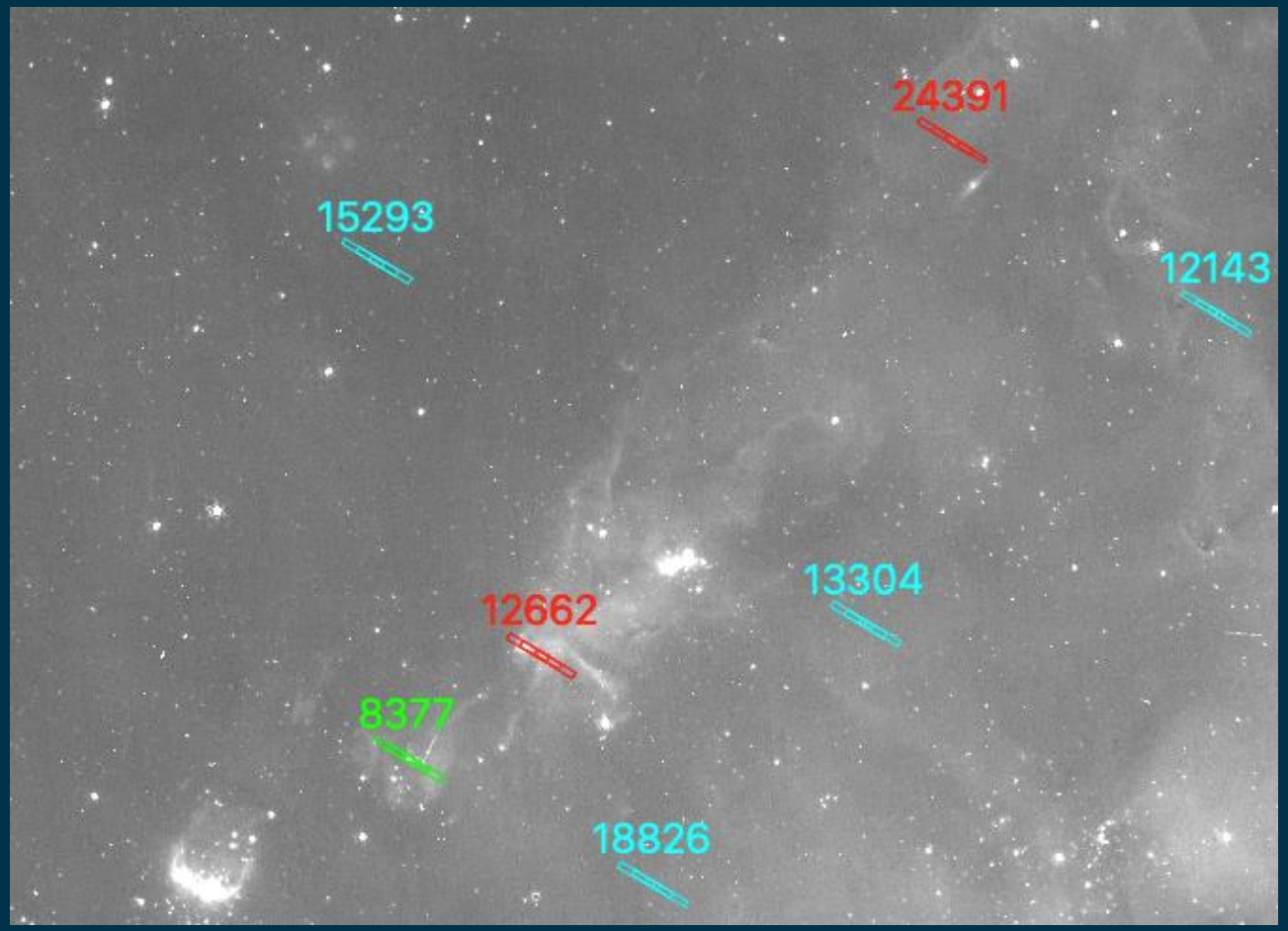
365x171x4 micro-shutters
~ 250,000

Tiny windows with shutters: 100x200 µm

- Works at cryogenic temperature (40K)
- Each shutter opens individually and wide



NIRSpec MOS observations of SF regions



NIRCam image F187N

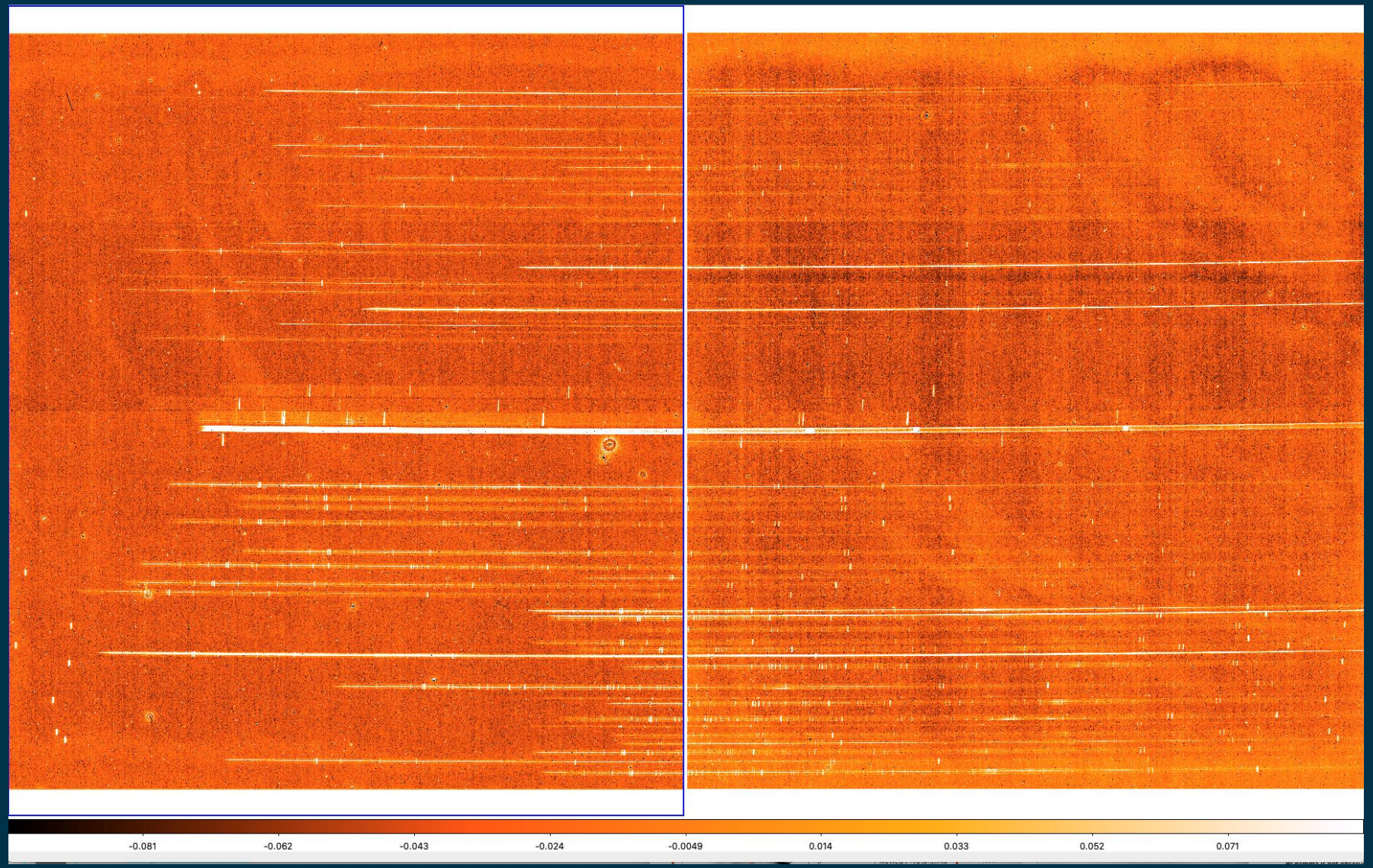
GTO program
led by Guido De Marchi
MOS Observations of two star-formation regions:

- NGC364 in SMC
- NGC3603 in our our galaxy

Excuted in July/August

Observation strategy:
3-shutters per targets
3-point nod
5-sky regions sampled over three exposures
➔ Standard approach for direct background subtraction

The data: exposures



MOS observations

- 40 stars
- Medium resolution spectroscopy F170LP-G235M
  $\Delta\lambda = 1.7 - 3.2\ \mu\mathrm{m}$

Reducing and analyzing NIRSpec data

Tasks:

- Run ESA team pre-processing pipeline (implemented in C): generates count-rate images from raw exposures (up-the-ramp integrations)
- Run ESA pipeline NIPS (python 3): generate flux- and wavelength calibrated spectra from detector traces
- Run python scripts to plot and analyze individual spectra

What else did we need?

What did we need?

- Access to the archived data in NIPS format
- A “private” place to work collaboratively on these non-public datasets, trying to avoid as much as possible to spread them all over the place.

➔ Remotely accessible platform

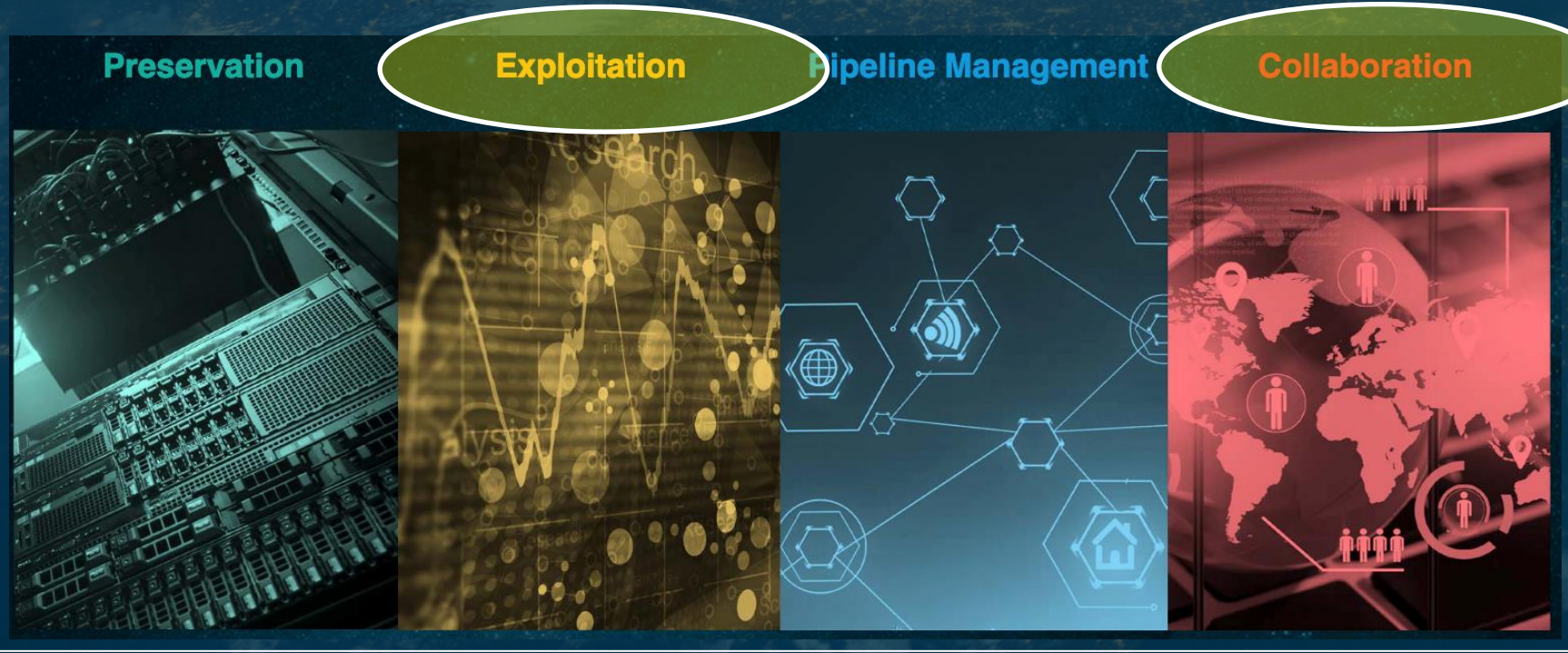
- ❖ Small team of 4 people running the data reductions based in Italy and the Netherlands
- ❖ The raw data from our observations become available in July and August, when we were on holiday and traveling to different places
- ❖  …. We needed a quick solution… a solution NOW…

ESA Datalabs entered the scene…

I had some experience of working with the Datalabs from the analysis of some parts of commissioning data that were conducted with external support from the GTO team - Tech talk by P. Ferruit

This was the right tool for our project!! 😅😌

Within a few days all of us could run the data reduction

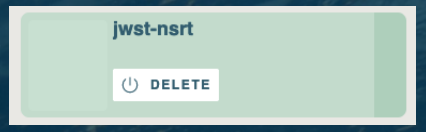


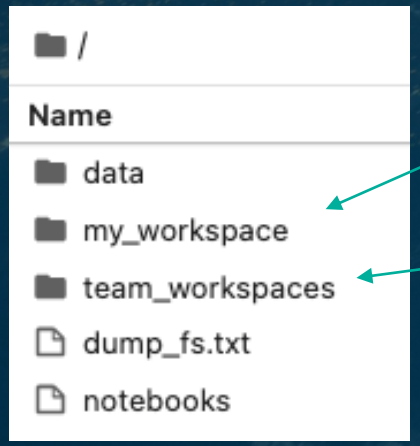
Note that we were not talking of "big data" in our case.

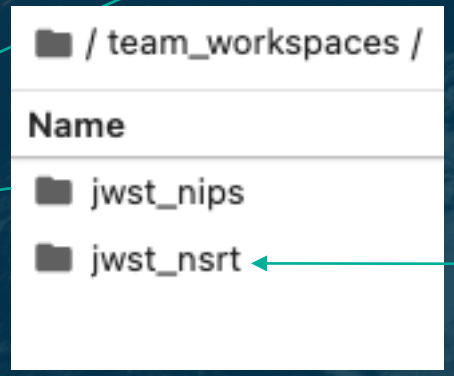
Accessing “private” work area in Datalabs

A shared NSRT volume was created only visible by the Datalabs users registered GTO users.

- Manual process during this development phase, expecting it to be “standard” in the released version.
- “Persistent” area, will survive the deletion of a jwst-nsrt Datalabs instance.
- Work perfomed by Datalabs team in collaboration with P. Ferruit
- Extensive support by Marcos





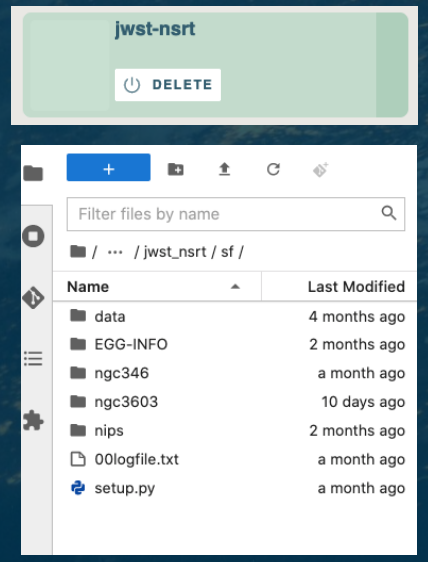


Standard folder in Datalabs.What users put here is only visible by themselves and is “persistent”. Not a shared space.

What users put here is only visible by NSRT users and is “persistent”. This is the shared space we requested.
(users belonging to multiple teams will see more folders)

Setting up our “private” Python environments

This was done by creating “local” environments stored in the shared volume. Their maintenance, configuration-control was our business.



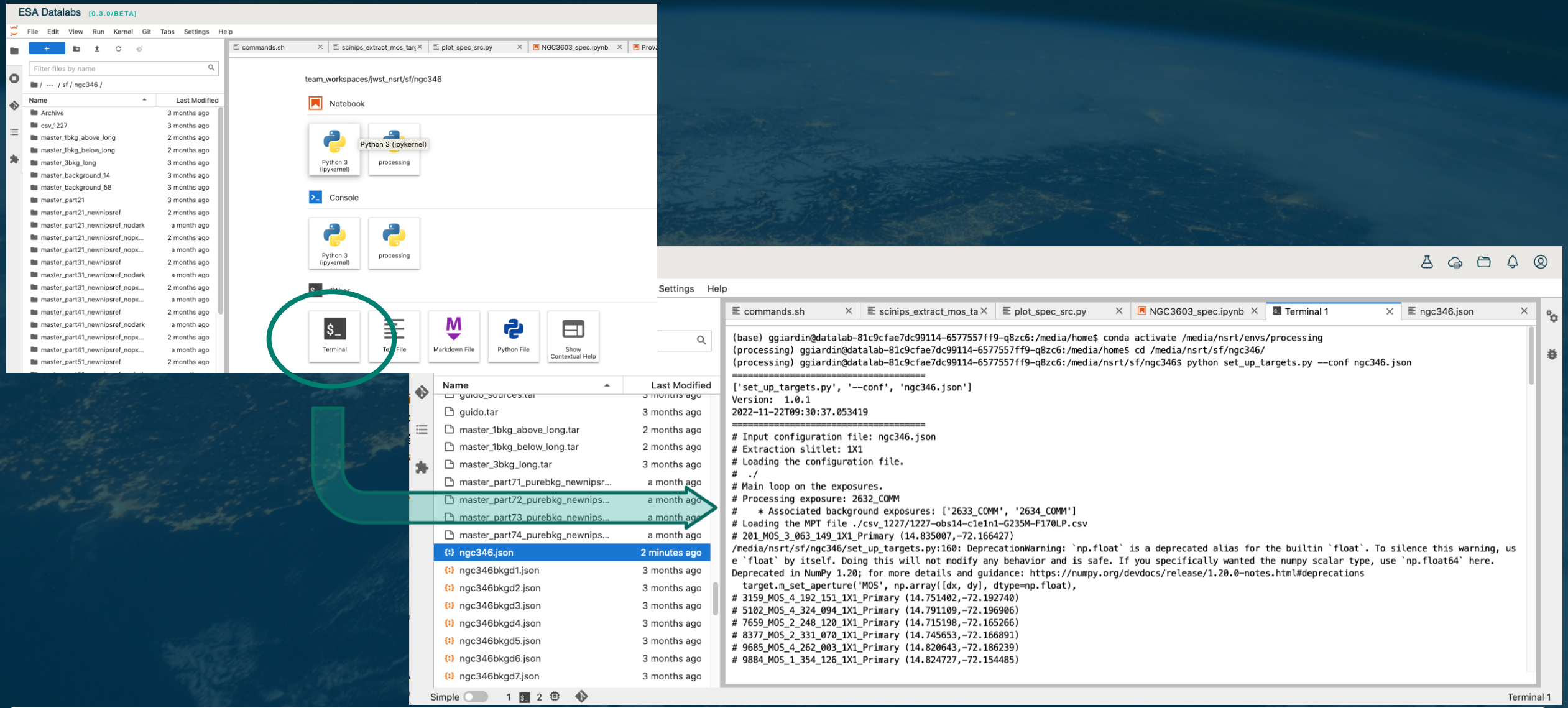
Specific call to the conda commands used to create (once!) and activate these environments.

```
55  # creating the trunk environment
56  conda create -p /media/nsrt/envs/processing python=3.9
57  conda activate /media/nsrt/envs/processing
```

With every new terminal sessions:

```
(base) ggiardin@datalab-81c9cfae7dc99114-6577557ff9-q8zc6:/media/home$ conda activate /media/nsrt/envs/processing
(processing) ggiardin@datalab-81c9cfae7dc99114-6577557ff9-q8zc6:/media/home$
(processing) ggiardin@datalab-81c9cfae7dc99114-6577557ff9-q8zc6:/media/home$ cd /media/nsrt/sf/ngc346/
```

Typical session: running python script in terminal



We also used Jupiter notebooks

commands. | scinips_extr | plot_spec_s | NGC3603_s | Prova.ipynb | Terminal 1 | Launcher | ngc346.jsor

Code | git | Python 3 (ipykernel)

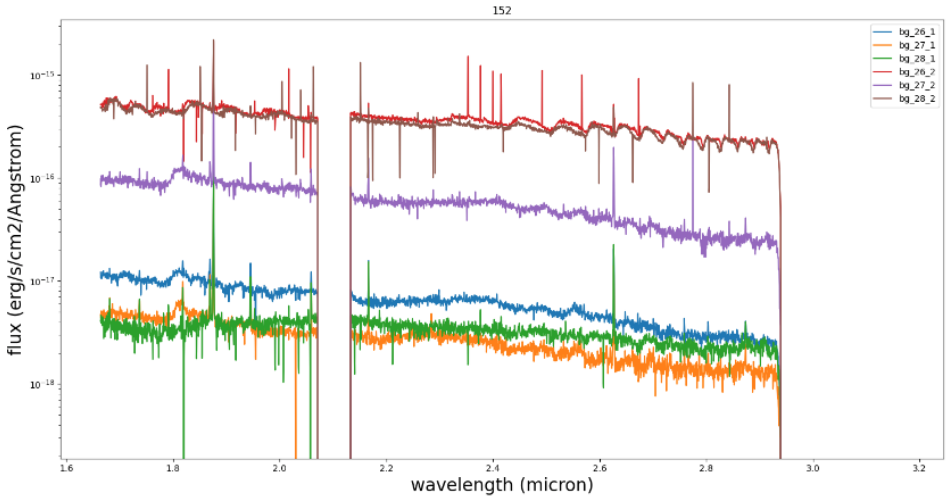
```
# ax.plot(wavelength_bg_28_3, flux_bg_28_3, label="bg_28")

# ax.plot(wavelength_no_pxbg, flux_no_pxbg, label="no_pxbg_20")
# ax.plot(wavelength_no_pxbg_21, flux_no_pxbg_21, label="no_pxbg_21")
# ax.plot(wavelength_no_pxbg_22, flux_no_pxbg_22, label="no_pxbg_22")

# for wave_count, wave in enumerate(lines_df["wavelength"]):
#     if 1 < wave < 5:
#         ax.vlines(wave, colors=colour_dict[lines_df["name"][wave_count]], ymin=0, ymax=np.nanmax(flux_bg_1)
#         ax.text(wave, 1.15*np.nanmax(flux_bg_1), spectral_lines[wave_count].split(" ")[0], color=colour_dic

# plt.ylim(10**-18, 10**-15)
# plt.xlim(1.66, 3.2)
plt.title(source_name)
ax.set_xlabel("wavelength (micron)", fontsize=20)
ax.set_ylabel("flux (erg/s/cm2/Angstrom)", fontsize=20)
ax.legend()
plt.yscale("log")
# plt.savefig(r"C:\Users\ciara\Documents\college\Leiden\nirspec\NGC3603_JWST\plots\NGC3603_1Ds_pxbg\1066_befo
plt.show()
```

Figure

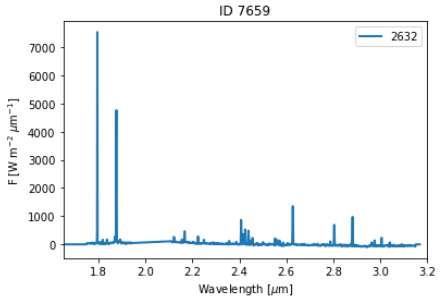


commands. | scinips_extr | plot_spec_s | NGC3603_s | Prova.ipynb | Terminal 1 | Launcher | ngc346.jsor

Code | git | processing

[20]:
```
%run plot_spec_src.py 7659 2632
```

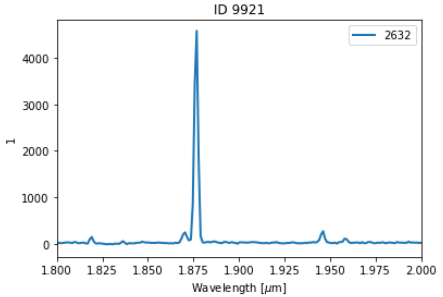
```
7659 7659_MOS_2_248_120_1X1_Primary
# Opening master/data/2632_COMM_F170LP_G235M_NOLAMP/2632_COMM_7659/products/2632_COMM_7659_EXTR_R1D.fits
WARNING: FITSFixedWarning: The WCS transformation has more axes (1) than the image it is associated with (0)
[astropy.wcs.wcs]
```



[ ]:
```
%matplotlib widget
```

[8]:
```
%run plot_spec_src.py 9921 2632
```

```
9921 9921_MOS_2_324_146_1X1_Primary
# Opening master_sources/data/2632_COMM_F170LP_G235M_NOLAMP/2632_COMM_9921/products/2632_COMM_9921_EXTR_R1D.f
its
WARNING: FITSFixedWarning: The WCS transformation has more axes (1) than the image it is associated with (0)
[astropy.wcs.wcs]
```



Data reduction in Python

What worked well:

- We were able to set up our work environment in Python without any problem. We were able to manage it by ourselves.
- Datalabs is very intuitive environment – in particular thanks to the “Terminal” application that allows one to work as one is used, on a standard computer running a Unix-like OS
- Speed!! NIPS pipeline is parallel software so having access to many cores meant we could re-run processing significantly faster than on our laptops: fast-turn around!
- Many re-processing due to criticalities of background subtraction…

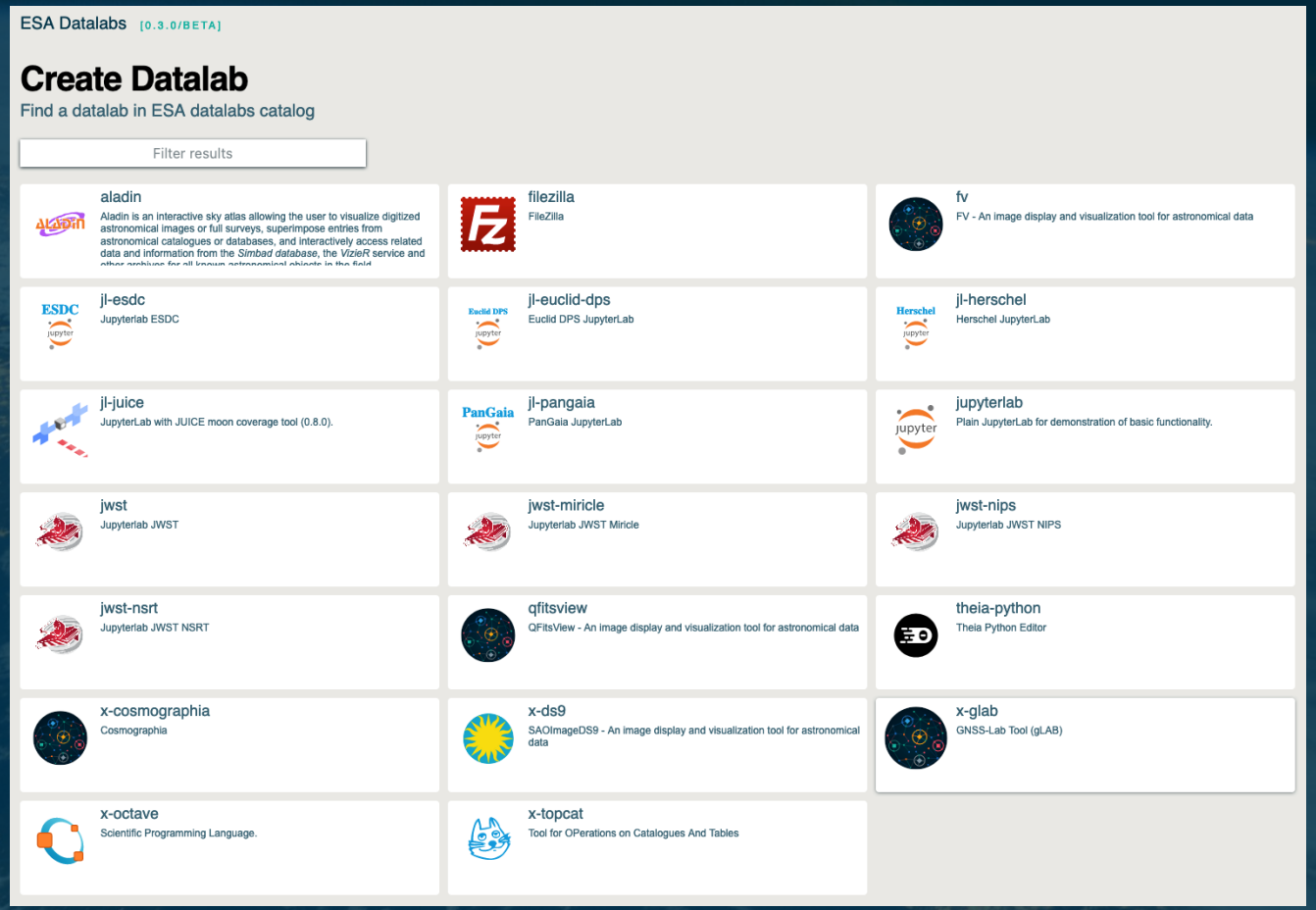
What did not work so well:

- Network… not a Datalabs-specific issue
- Visualisation

Adapting our way of working to Datalabs

Datalabs is a notebook-oriented environment if one wants to display things, when using python, notebook is the way to go

Generally, one cannot use applications with GUIs in Datalabs unless they are available in the Datalabs catalog.



→ This point is actively worked on by the datalabs team.

Conclusions

- We needed a platform to work collaboratively on reducing and analyzing NIRSpec data using specific software: ESA team data reduction software

- We found Datalabs a stable and intuitive environment for us to efficiently process the data and share the high level products

- Scientific analysis of these products is on going (lines, their intensities, ratios, etc.)… soon to be revealed!
-