Data reduction and analysis of FIFI-LS data

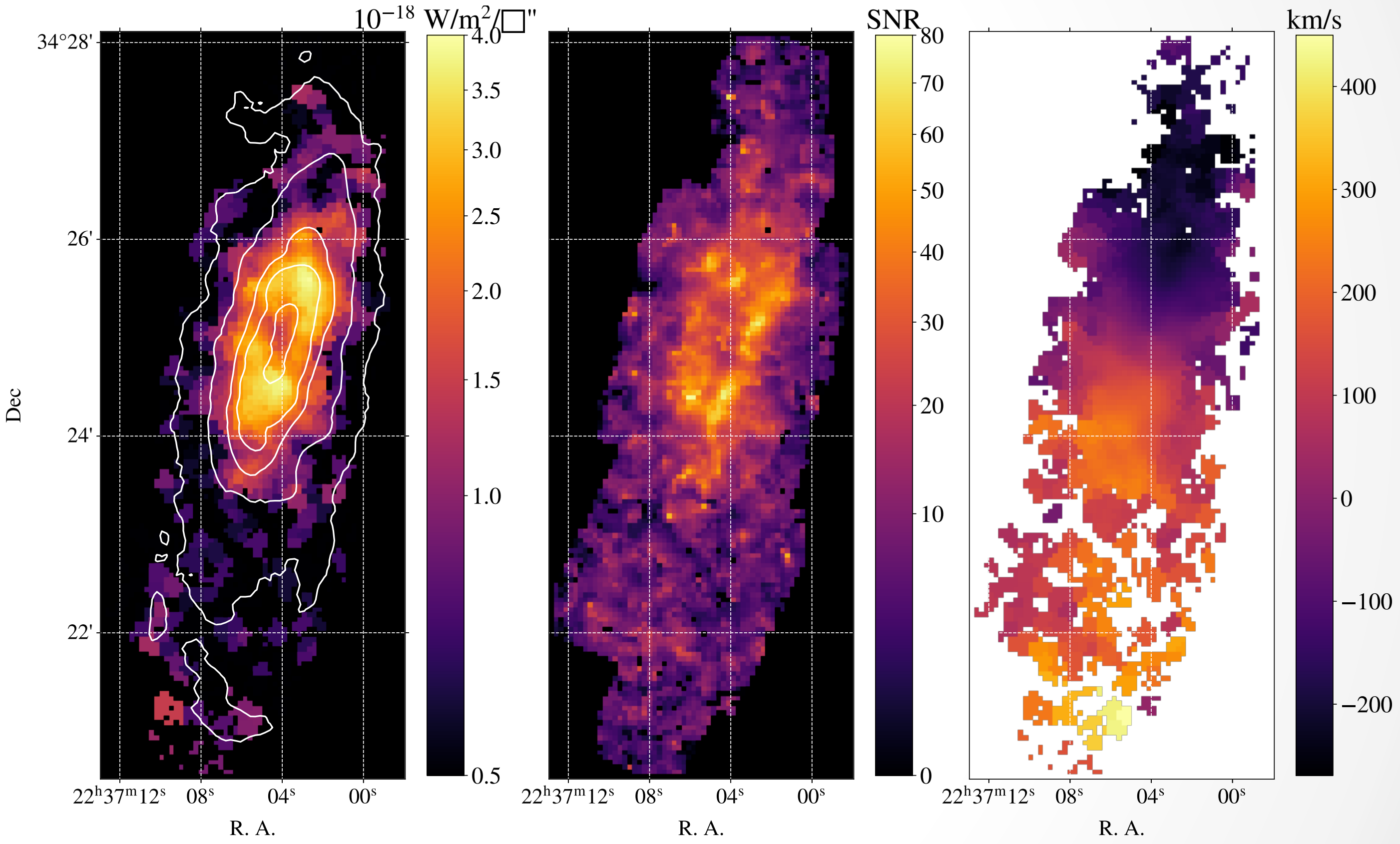
Dario Fadda
SOFIA Science Center

Dario Fadda
SOFIA data school - 16 Apr 2023

Outline

We will go through three essential steps with FIFI-LS data:

1. Download
2. Reduce
3. Analyze



Getting the data: selection

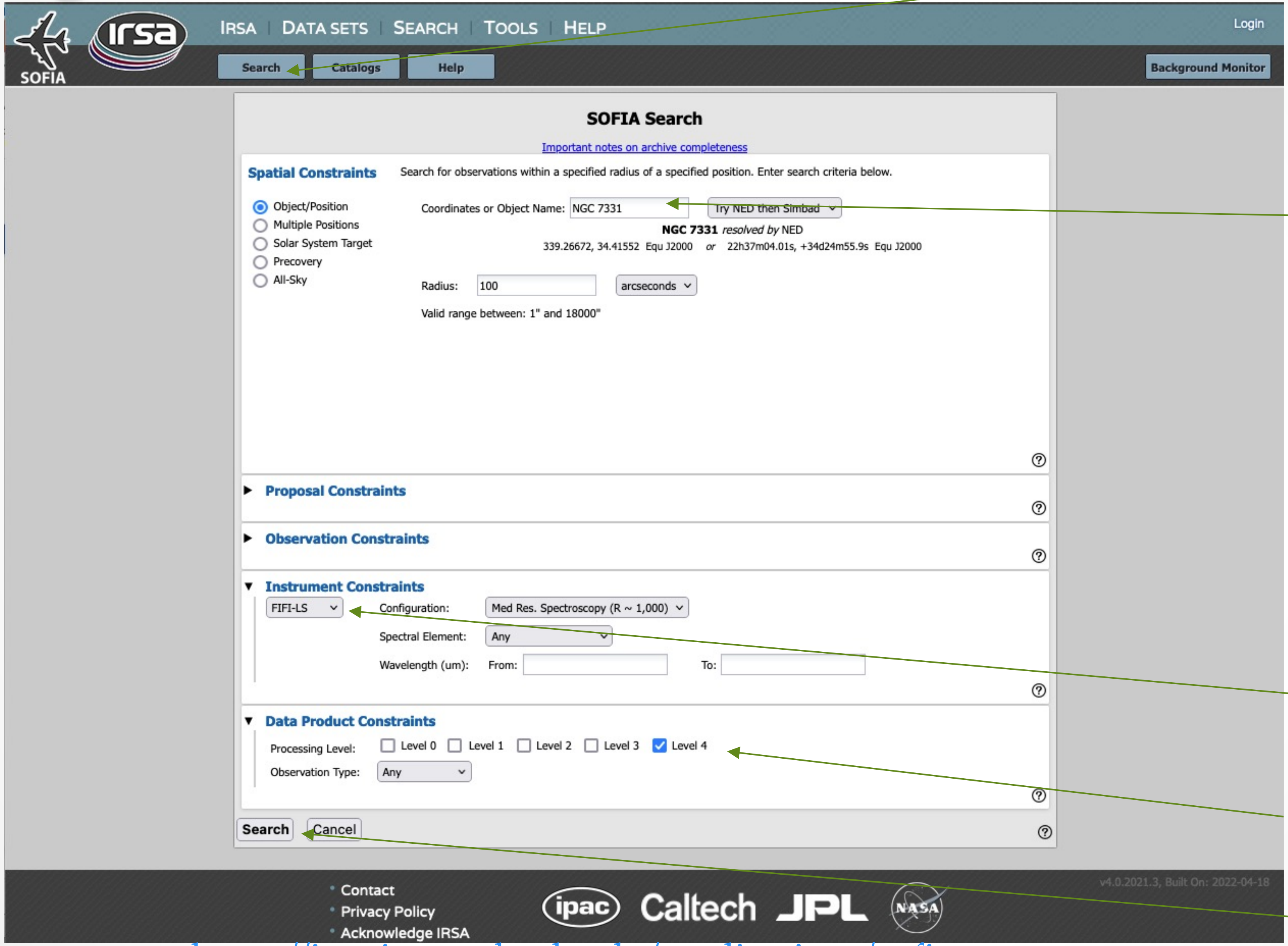
Search tab

Object name

Instrument

Processing level

Search button



https://irsa.ipac.caltech.edu/applications/sofia

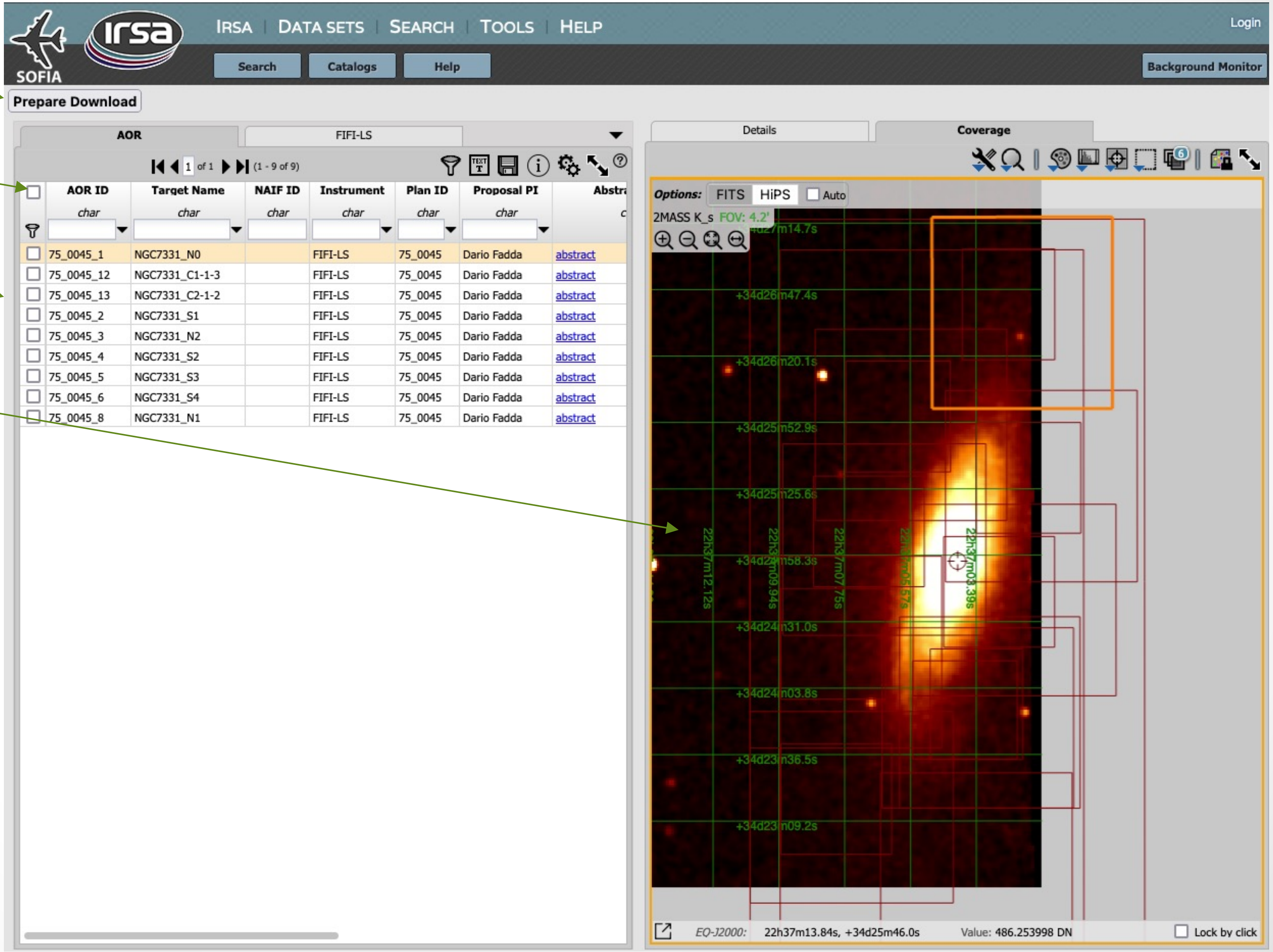
Getting the data: download

Download

Select AORs

Different observations

Coverage



Dario Fadda
SOFIA data school - 16 Apr 2023

Raw data only

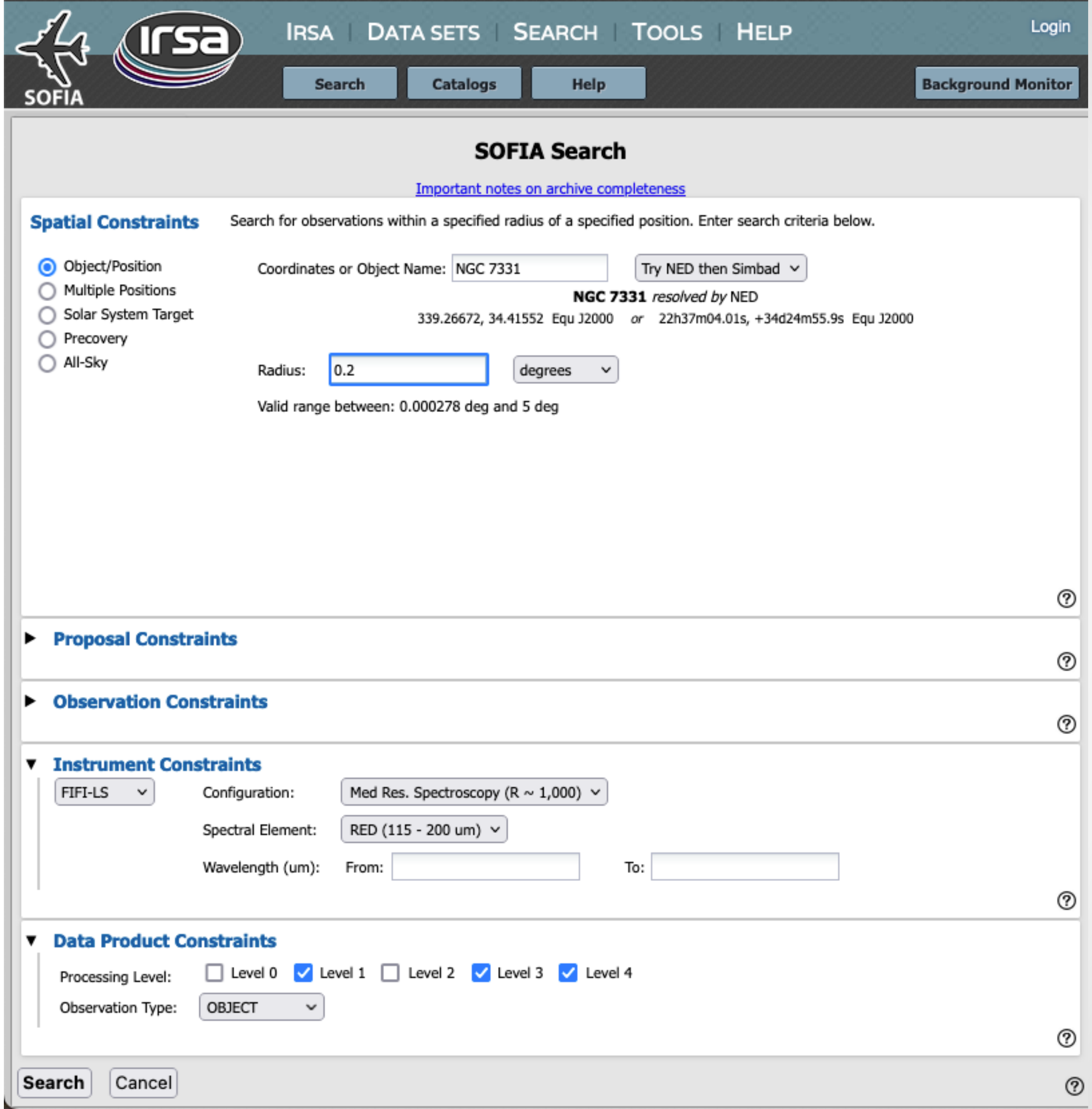
To download raw data only:

- Select Level 1
- Observation type: OBJECT

Important:
Use a radius larger than that suggested (0.2 degs)

Select:

- Instrument: FIFI-LS
- Spectral element: Red
  (to download [CII] obs only)

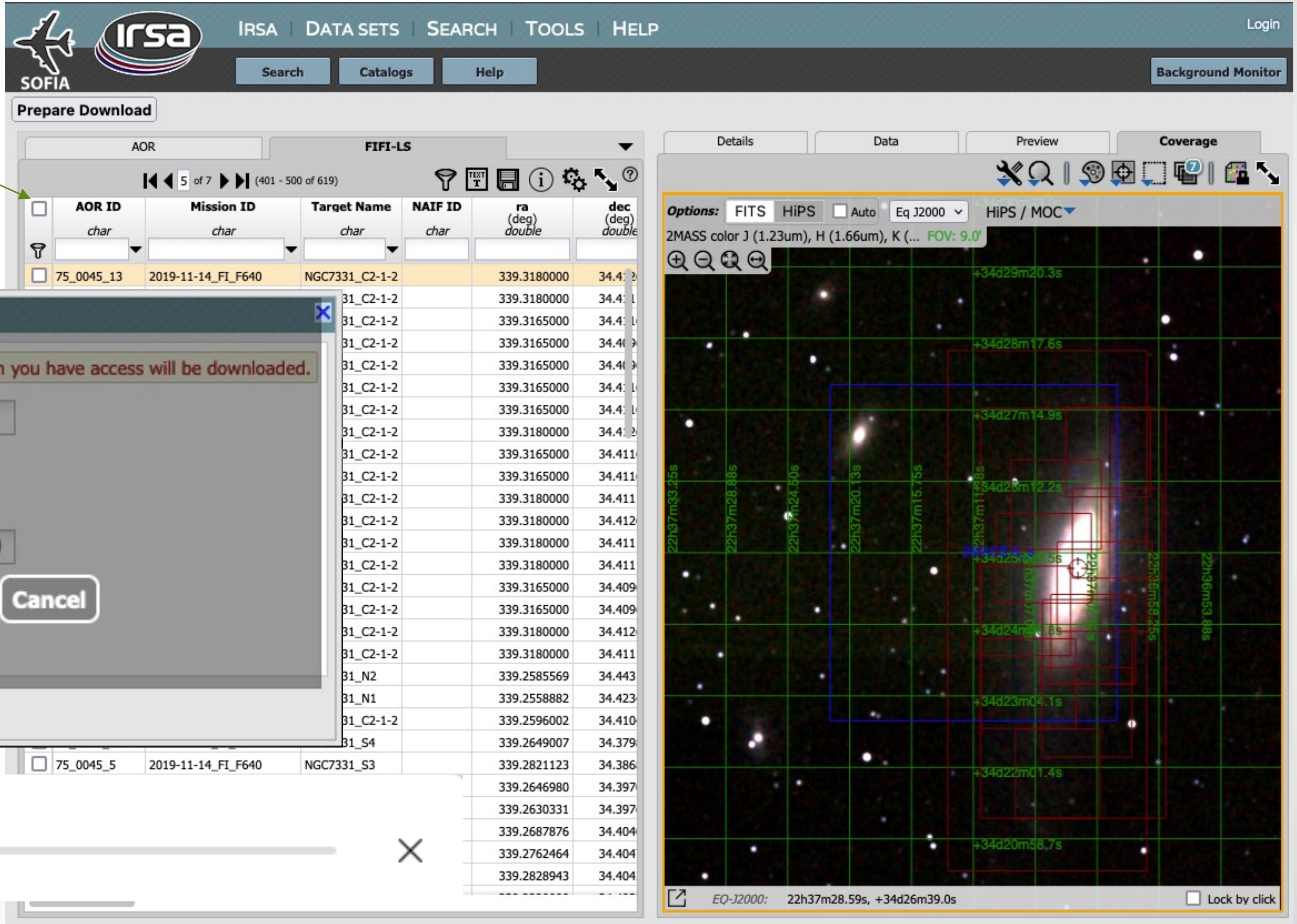


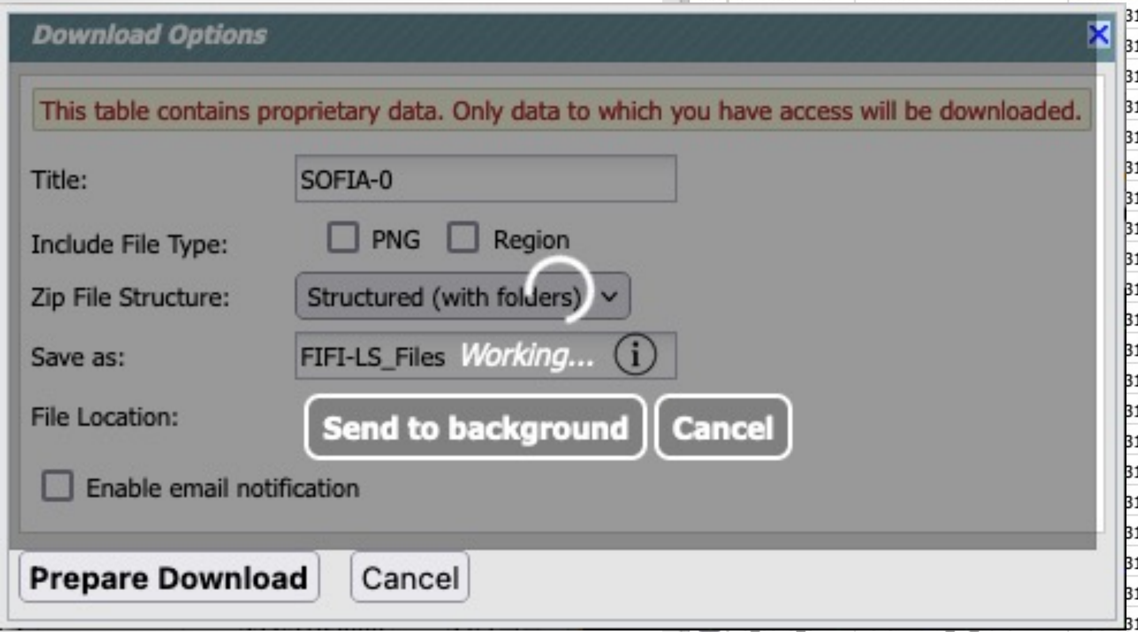
Dario Fadda
SOFIA data school - 16 Apr 2023

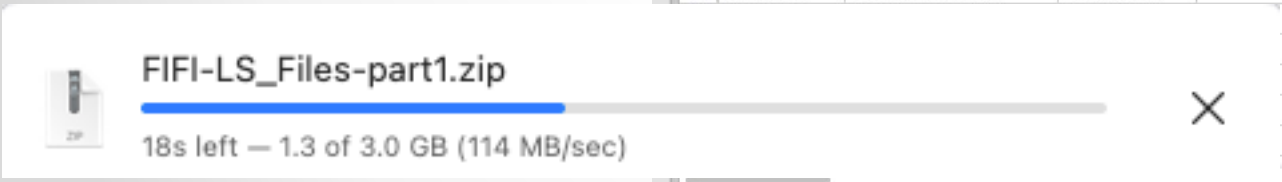
Select and download

Select all AORs.
Prepare download.

Skip the PNG to download less stuff.

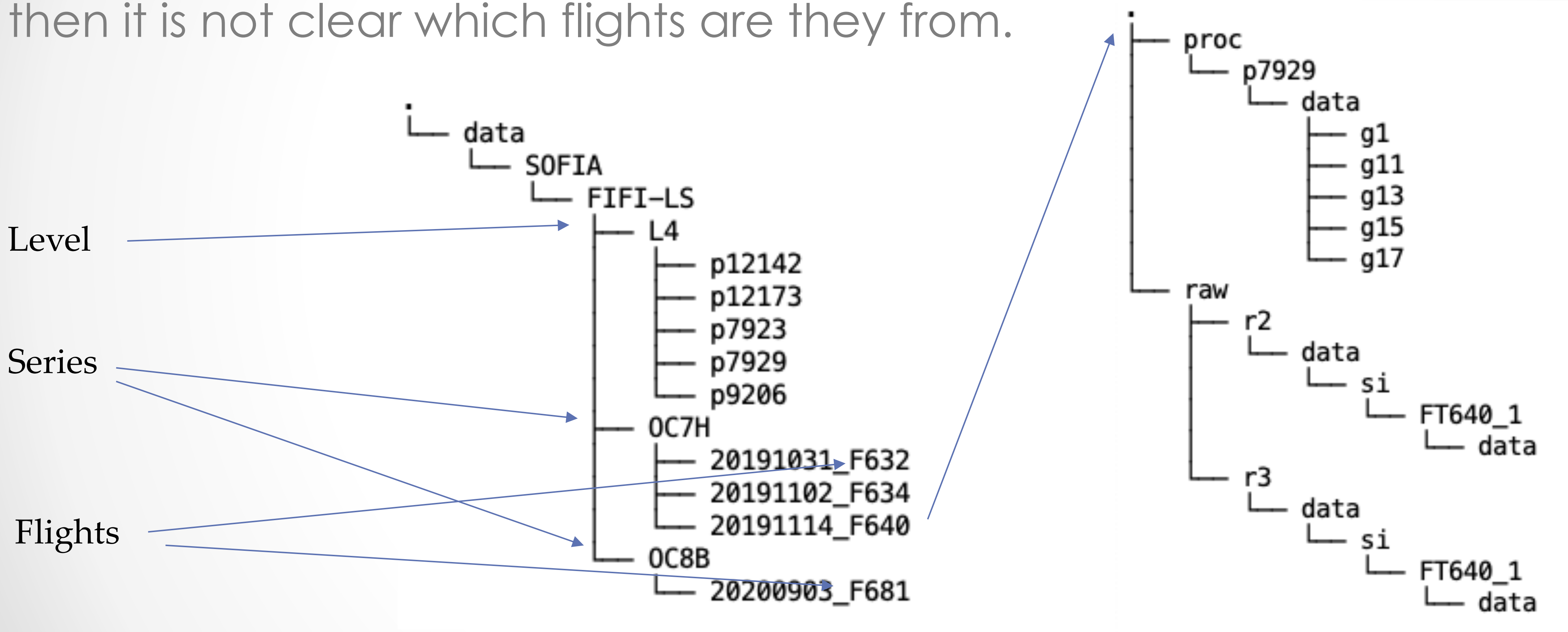






Exploring the data tree

In this case, the whole set of data is 3.0 GB. We are interested in the raw files only. It is possible to get all the files in one directory only, but then it is not clear which flights are they from.



Each flights contains processed and raw data directories. For our reduction we will consider only raw data.

Dario Fadda

USRA DLR DSI NASA SOFIA

Raw files

Raw files from the archive have some information in their names:

```
00415_072207_00013_NGC7331_S2_act4_-2_-5_A_lw.fits
00416_072245_00014_NGC7331_S2_act4_-2_-5_B_lw.fits
```

Progressive run number | UTC time acquisition | Observation number | Target name | Part of observation | Offsets | Nod | Channel

Each file contains the readouts for a certain number of grating positions for one channel only.

For each telescope pointing there is a new file.

Why using the pipeline ?

The IRSA archive contains level 4 data ready to be used for science.

Nevertheless, there are advantages to use the pipeline to reduce the raw data:

- The processing of data has been done in different times with different versions of the pipeline and of calibration (flats, flux)
- You can be interested in combining data of the same object from different series or bigger regions which are reduced in pieces
- You would like to selectively reject some files (which have high noise), or you want to resample the final data with a different smoothing parameter

Installing the pipeline

- Documentation: https://sofia-usra.github.io/sofia_redux/index.html
- Repository: https://github.com/SOFIA-USRA/sofia_redux
- Installation: https://github.com/SOFIA-USRA/sofia_redux/blob/main/docs/install.rst

Suggested installation:

- Install Anaconda Python: https://www.anaconda.com/products/distribution
- Download the environment file: https://raw.githubusercontent.com/SOFIA-USRA/sofia_redux/main/environment.yml
- Create and activate the pipeline environment:
  conda env create -f environment.yml
  conda activate sofia_redux
- Clone the repository:
  git clone https://github.com/SOFIA-USRA/sofia_redux.git
- Install in edit mode:
  pip install –e sofia_redux

Updating the pipeline

Before starting any new reduction, it is a good habit to update your local repository in the case there is a new version of the pipeline.

If you cloned the repository, just go inside it and pull the new version.

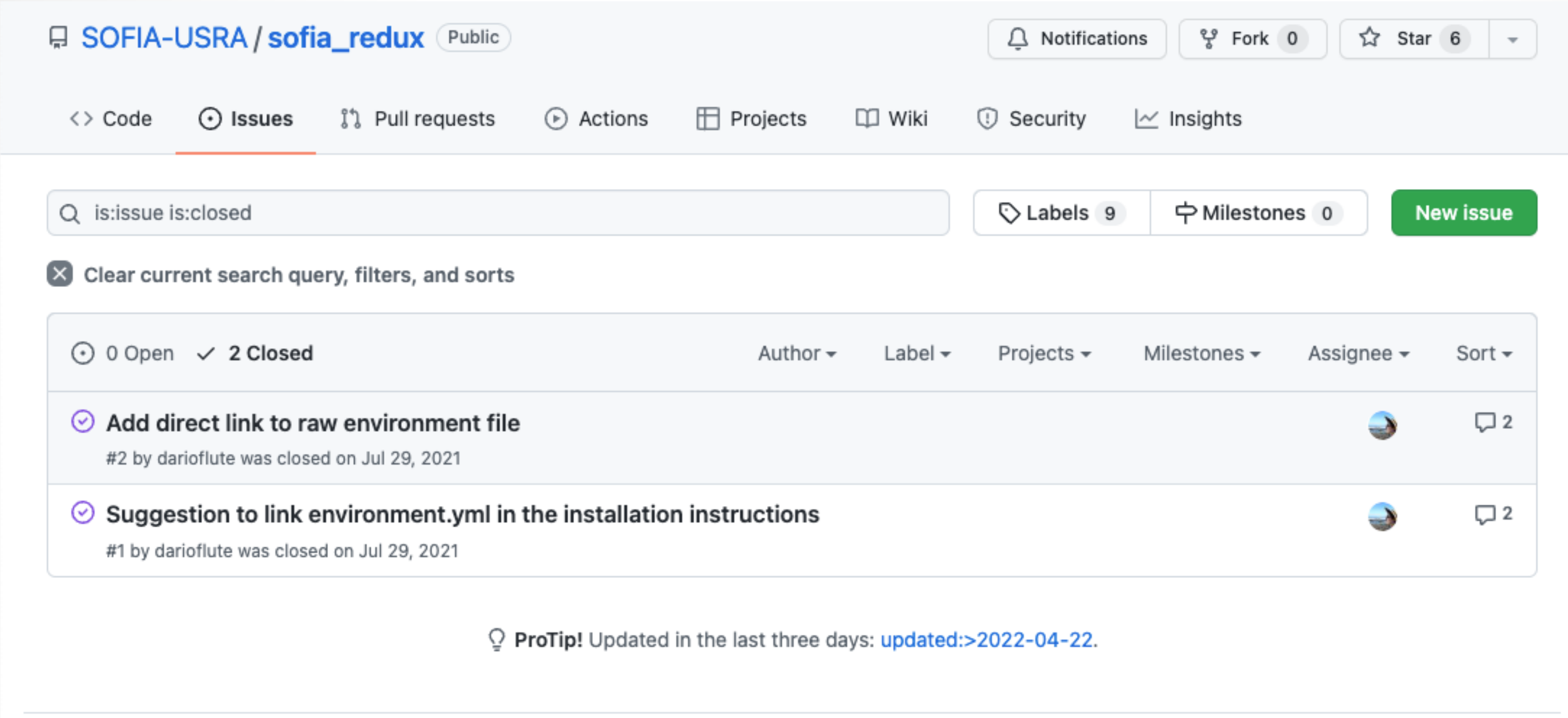
```
cd sofia_redux
git pull
```

Since you installed the pipeline in ‘edit’ mode, the new pipeline will be immediately activated.

If you want to try the pipeline, you can opt for a direct conda installation:

```
conda install -c sofia-usra -c astropy sofia_redux
```

Contributing to the pipeline



Any issue, comment, or proposal for new features can be written in the github page of sofia_redux.
You will need a github account (you can open it for free). Once logged in, click on the 'Issues' tab.
At this point, this window will appear showing open and closed issues.
You can open a new issue by clicking on the green button.

Adding the library of atmospheric models

- The library of atmospheric models (ATRAN) should be downloaded separately through the URL:

  https://sofia-downloads.s3-us-gov-west-1.amazonaws.com/atran_fifi-ls_wv.tgz

This can be unarchived in a directory which will be then referenced in the parameter file used for the reduction.

Explore data through a Python notebook

The raw files store all the ramps of each observation. It is possible to look into the details of the data before starting the reduction.

This can help individuating files to reject, especially those corrupted.

A companion Jupyter notebook can help with this exploration.

The notebook can be uploaded from:

https://www.sofia.usra.edu/sites/default/files/2022-04/FIFI-LS.ipynb

Running the pipeline

- The pipeline is written in sequential modules
- Each module has options
- Products from each step can be saved and inspected
- Some modules can be run in multi-threading to speed them up

The pipeline can be run via a GUI (redux).

See this presentation to know how:

https://www.sofia.usra.edu/meetings-and-events/events/pipeline-week-introduction-sofia-processing-pipeline

Logo of the REDUX pipeline

I will show how to run it through a command line by specifying options and modules to run in a configuration file.

Running the pipeline

After entering the sofia_redux environment:

```
conda activate sofia_redux
```

You will need a configuration file and the list of input files.
Once these are created, you can run the pipeline with:

```
redux_pipe –c redux_param.cfg redux_infiles.txt
```

Where redux_param.cfg is the configuration file and redux_infiles.txt contains the list of the input files.

Dario Fadda
SOFIA data school - 16 Apr 2023

The list of input files

To create a list of files, after unzipping the data, a directory called 'data' is created and in the same path one can create another directory to reduce the data.

First, one need to create a list of files:

```
mkdir reduced
cd reduced
find ../data –name '*lw.fits' > redux_infiles.txt
```

Then, one needs a configuration file with all the parameters needed for the reduction: redux_param.cfg

Pipeline configuration: basic modules

Let's examine the parameters contained in the redux_param.cfg file

```
# Redux parameters for FIFI-LS instrument in IFS mode
# Pipeline: FIFI_LS_REDUX v2_6_1_dev0
recipe = checkhead, split_grating_and_chop, fit_ramps
[1: checkhead]
    abort = True
[2: split_grating_and_chop]
    save = False
[3: fit_ramps]
    save = False
    parallel = True
    s2n = 10.0
    thresh = 5.0
    badpix_file = ""
    remove_first = True
    subtract_bias = True
    indpos_sigma = 3.0
[4: subtract_chops]
    save = False
[5: combine_nods]
    save = False
    b_nod_method = nearest
    offbeam = False
```

List of module to run (optional)

Checking headers

Separate grating and chop positions

Slope of the ramps

Subtract sky from source signal

Combine the two nods

Dario Fadda
SOFIA data school - 16 Apr 2023

Pipeline configuration: calibration

```
[6: lambda_calibrate]
    save = False
[7: spatial_calibrate]
    save = False
    rotate = True
    flipsign = default
[8: apply_static_flat]
    save = False
    skip_flat = False
    skip_err = True
[9: combine_grating_scans]
    save = False
    bias = True
```

Apply wavelength calibration

Add astrometry

Apply NE orientation

Apply flats

Do not apply flat error to error propagation

Combine different grating scans in a spectrum

A part deciding to save the files, the only parameter we should mention is the possibility to rotate the field in the usual NE direction or to leave it along the array coordinates. This is done with rotate.

Pipeline conf: tellurics & flux calibration

```
[10: telluric_correct]
    save = False
    skip_tell = False
    atran_dir = ""
    cutoff = 0.6
    use_wv = False
[11: flux_calibrate]
    save = False
    skip_cal = False
    response_file = ""
[12: correct_wave_shift]
    save = True
    skip_shift = False
```

Atmospheric correction

Path to ATRAN models

Cutoff in atmospheric transmission

Calibration in flux

Barycentric correction

Atmospheric correction with 'use_wv' = True is now possible for all the data, since we archived values of the precipitated water vapor for all the observations.
The cutoff is the level under which data should not be considered since the atmospheric transmission is too low. It is safe to put it at 0.6.

Pipeline: cube resampling

The resampling step put all the information together into a spectral cube with 2 spatial and 1 spectral dimensions.

```
[13: resample]
     save = True
     parallel = True
     max_cores = ""
     check_memory = True
     skip_coadd = False
     interpolate = False
     error_weighting = True
     fitthresh = -1
     posthresh = -1
     negthresh = -1
     append_weights = False
     skip_uncorrected = False
     scan_reduction = False
     save_scan = False
     scan_options = ""
     detector_coordinates = False
     xy_oversample = 5.0
     xy_pixel_size = 3.0
     xy_order = 2
     xy_window = 2.0
     xy_smoothing = 0.5
     xy_edge_threshold = 0.5
     adaptive_algorithm = none
     w_oversample = 8.0
     w_pixel_size = ""
     w_order = 2
     w_window = 0.8
     w_smoothing = 0.3
     w_edge_threshold = 0.5
```

general

spatial

spectral

Parameters for multi-threading

Fluxes weighted by variance

Rejection not activated

Parameters for OTF mapping

Oversampling
Output pixel size (arcsec)
Order of polynomial fit
Size of window and kernel for spatial smoothing
Size for limiting border effects
Options: none, scaled, shaped

Same meanings for the spectral dimension

Running the pipeline

Once all the parameters are chosen and the list of files is defined, once in the sofia_redux environments, one can lunch the reduction with the command:

redux_pipe –c redux_param.cfg redux_infiles.txt

If all the files are usable, the pipeline will run through all the modules and write the final spectral cube as well as any intermediate products which were required.

We saved the single frames before resampling them into the cube to show where they are taken.

Notes about resampling

The resampling involves some choices.

Since the data taken are sparse, the smoothing allows one to have a continuous cube across space and wavelengths.

Spatially, the size of FIFI-LS pixels are not much smaller than the PSF. This means that dithering is used to recover the PSF.

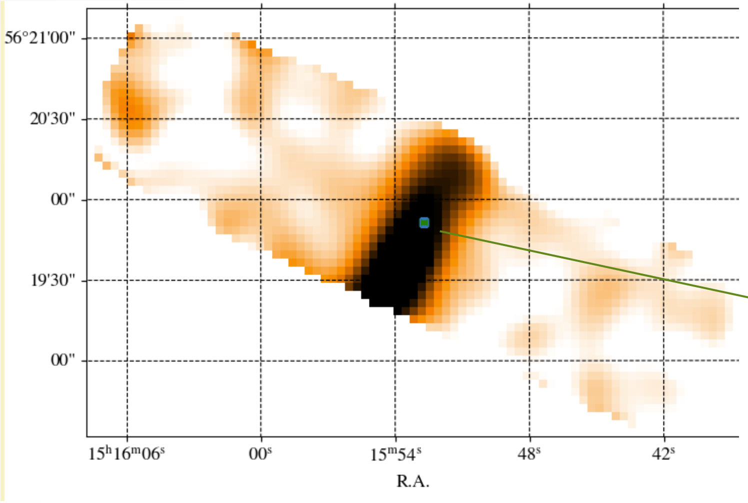
Clearly, over-smoothing will return a cube with broad point sources.

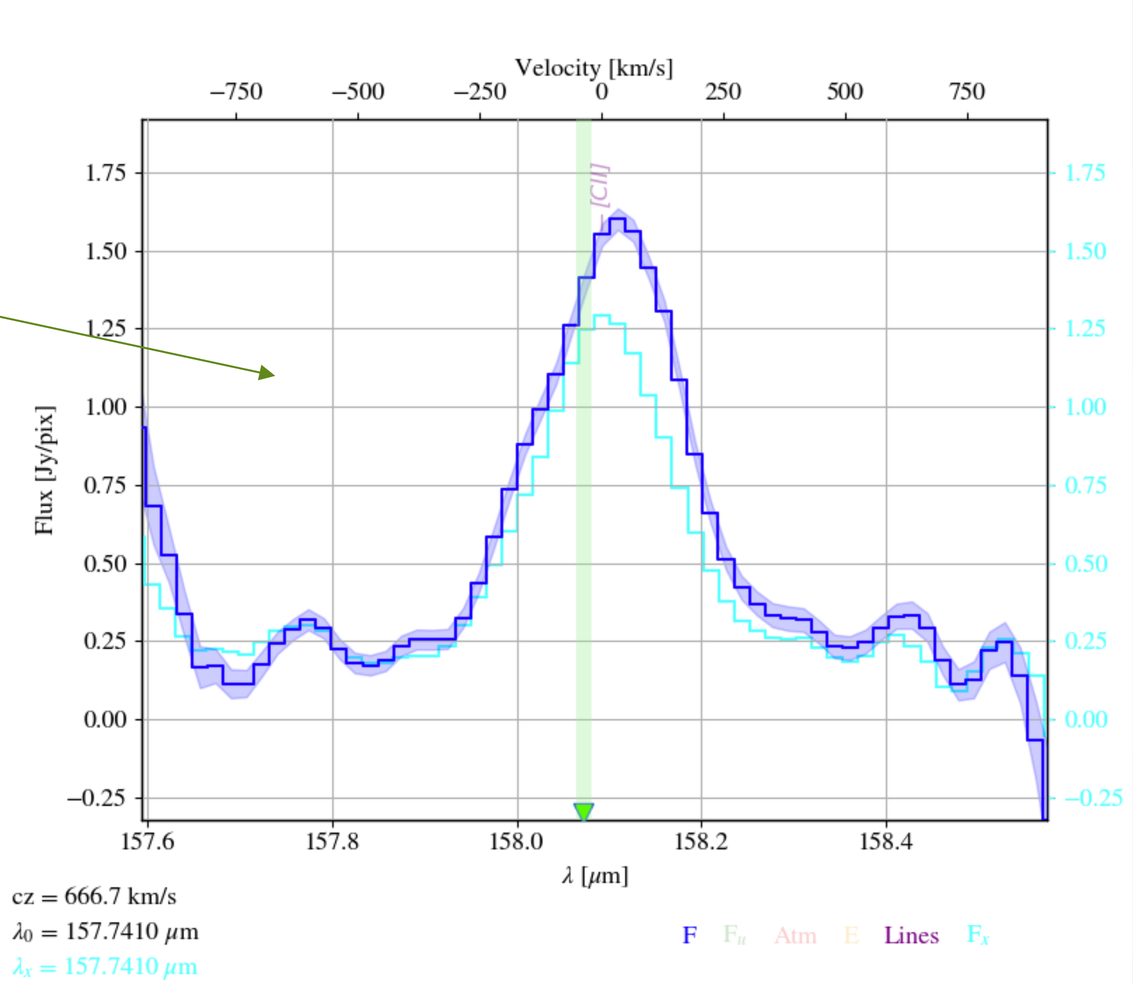
The peak flux on the top of the source will be depressed.

On the other hand, too little smoothing returns a noisy cube.

If the field contains diffuse emission and point sources, a solution is applying the 'adaptive algorithm' where the kernel is not fixed all over the cube but varies according to the flux. This is not used for the archival products.

Effects of over-smoothing





The danger of over-smoothing peaks of emission is clear along the center of the edge galaxy NGC 5907.

By choosing a kernel with a width corresponding to the FWHM of the spatial PSF, there is a loss of flux in the central ridge of the galaxy. This means that apertures corresponding to the PSF will contain less flux than what observed.

The choice of an adaptive kernel in this case leads to a smoothing very close to the blue curve, which has smoothing = 0.5 x FWHM.

Blue: xy_smoothing 0.5. (conserves spatial resolution)
Cyan: xy_smoothing 1.0 (oversmoothed)

Pipeline products

Several files are produced by the pipeline.
Each file has a different label indicating its type. Some useful files are listed here:

- *_RP[01]_*.fits results of ramp fitting (on and off positions)
- *_WAV_*.fits wavelength calibrated files
- *_XYC_*.fits spatially calibrated files
- *_FLF_*.fits flat-fielded files
- *_SCM_*.fits combined scans files
- *_CAL_*.fits flux calibrated files
- *_WSH_*.fits barycentric correction applied to wavelengths
- *_WXY_*.fits final cube resampled

WXY files contain several extensions. E.g., they contain also the cubes before applying the atmospheric corrections.

Exploring the data

I developed two codes to explore the data.
The codes are in Python and use the anaconda distribution.
To use them, you can create a conda environment and install them.

```
conda create -n fifils python=3.10
conda activate fifils
conda install -c conda-forge lmfit reproject fitsio
conda install -c darioflute sospex fifipy
```

I created versions for linux, windows, and the two flavors of mac-osx with the latest version of anaconda which uses Python 3.10

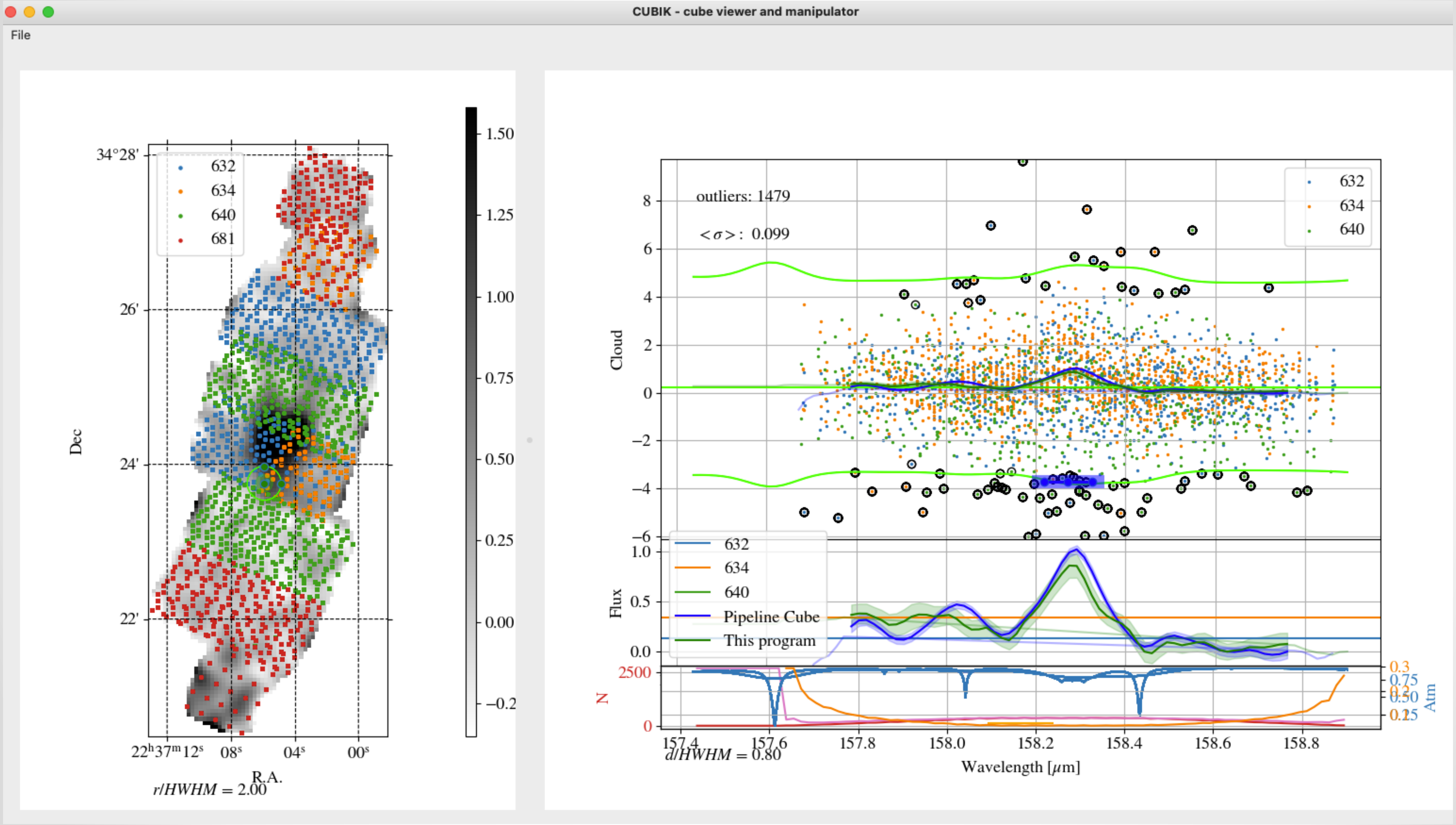
Explore the resampling with cubik

To explore the data used to obtain the resample cube and to have a better sense of the kernels used for the resampling, one can use cubik which comes with the library fifipy.

Lunch from a terminal: cubik

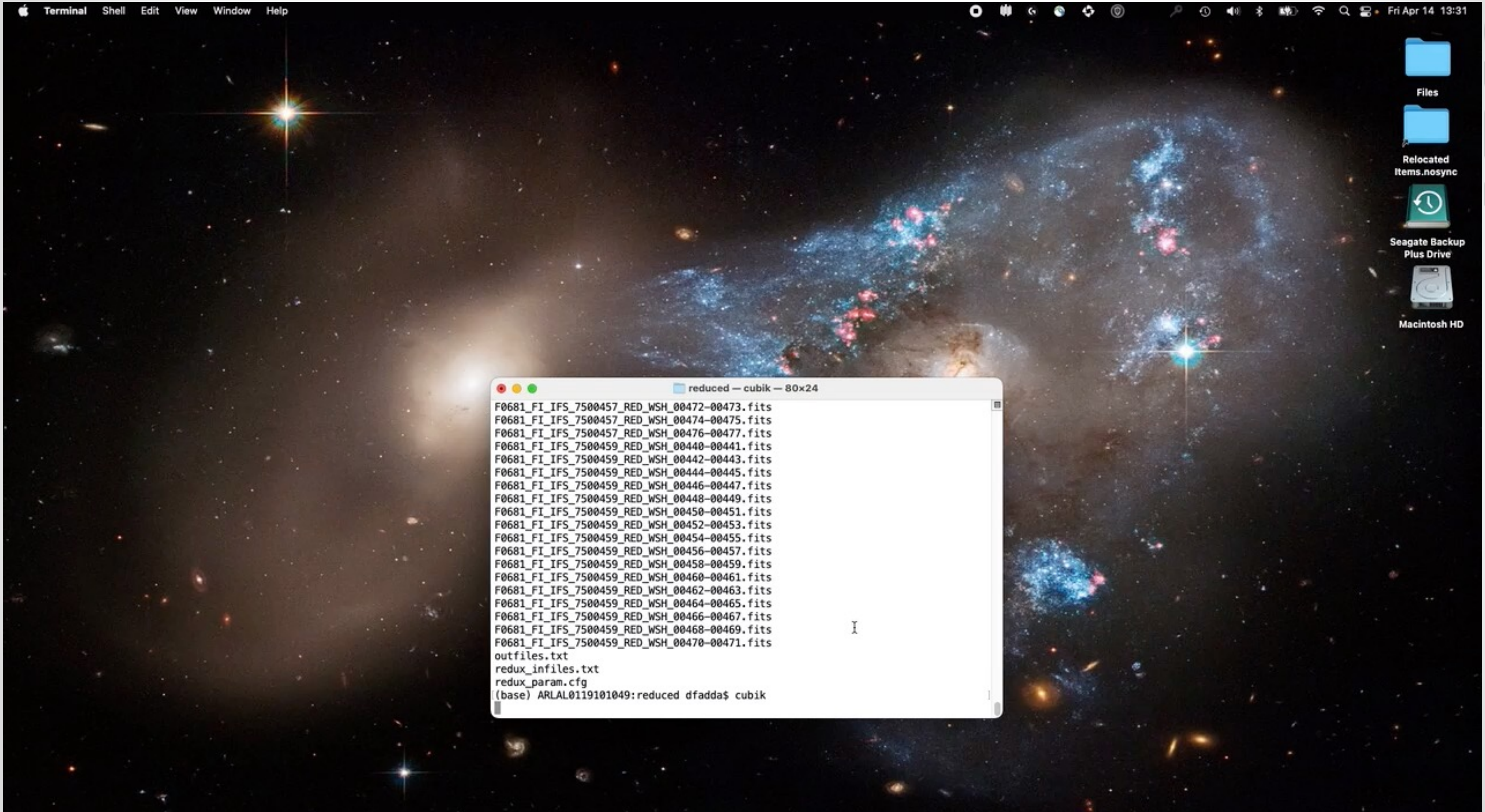
Then select the *WXY* file, the spectral cube. To work, the files *WSH* should be present in the same directory.

The code will show the spatial and spectral kernels used to resample the data as well as the cloud of points used for the resample which are included in the circular aperture on the left panel. The pipeline spectrum is shown overlapped on the cubik resampled data. Cubik can create a new cube or simply a new error cube based on the distribution of the data.



Dario Fadda
SOFIA data school - 16 Apr 2023

Cubik: the movie



Analyze the cube with sospex

Spectral cubes can be easily analyzed using sospex, a Python GUI which can be installed with anaconda:

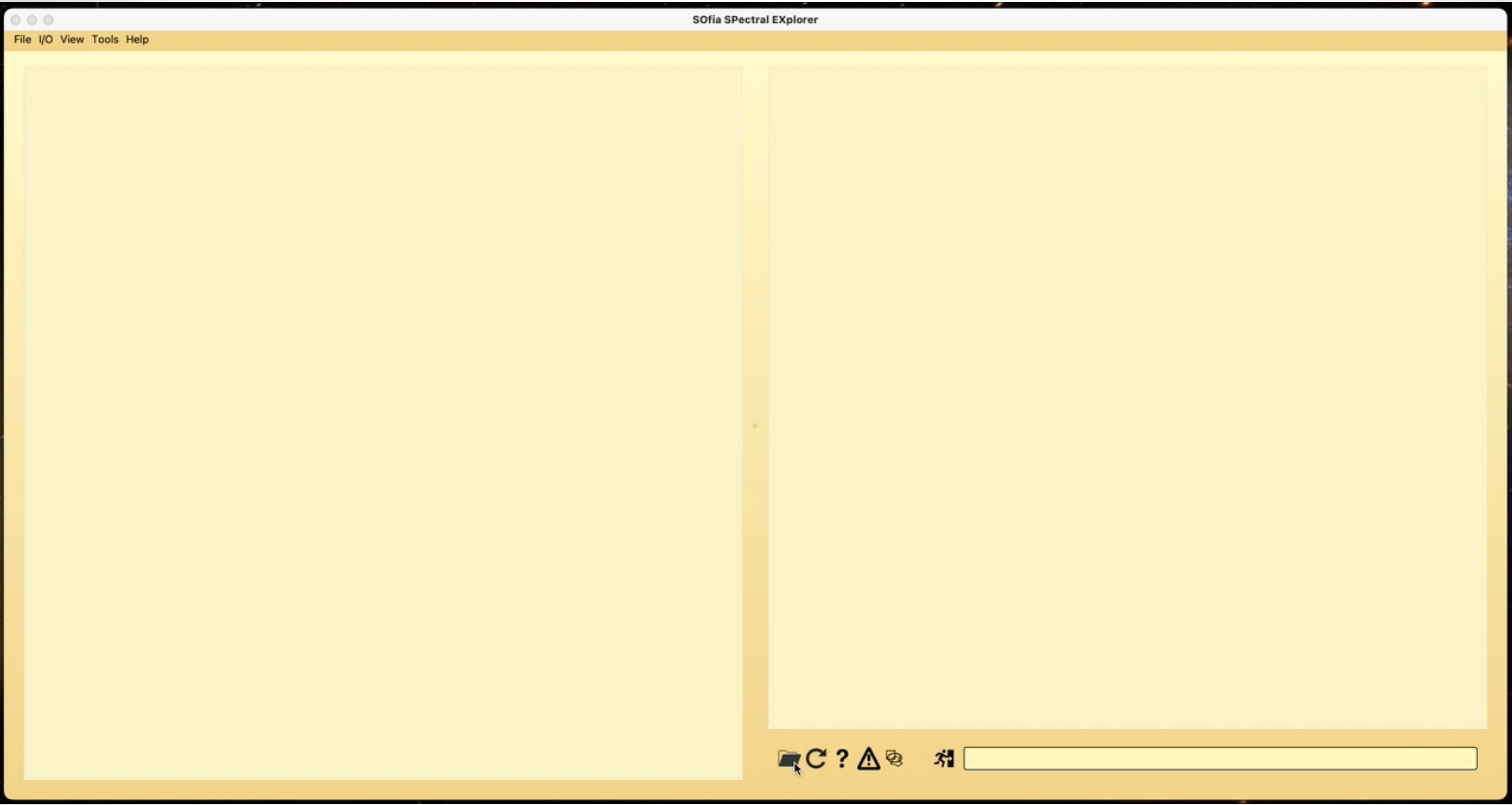
- https://github.com/darioflute/sospex/blob/master/INSTALL.md

The github repository is:

- https://github.com/darioflute/sospex

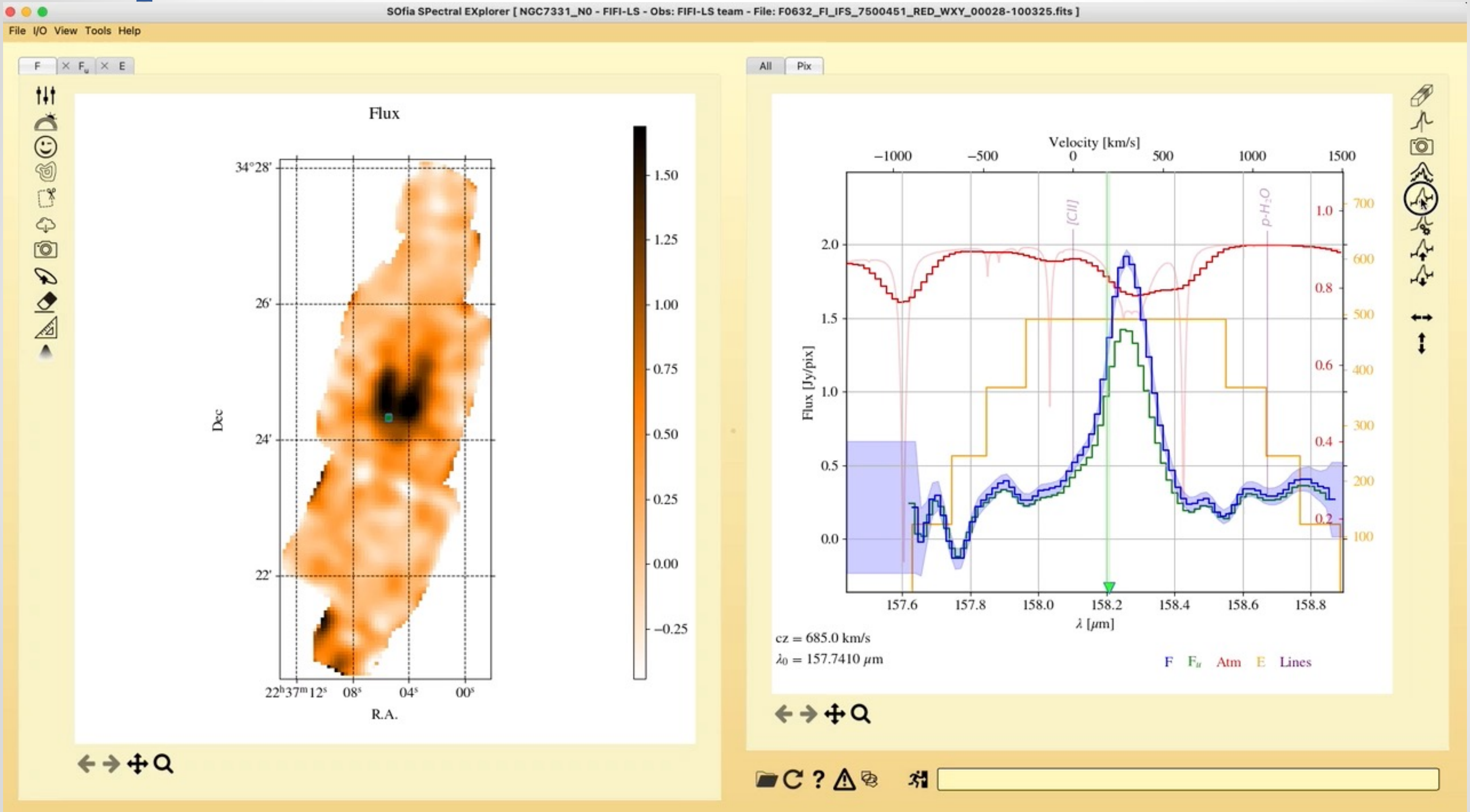
The GUI allows one to explore the cube produced by the pipeline and to do scientific analysis of the data (extract apertures, fit one or more lines, fit the entire cube, compare with other images/spectral cubes).

sospex: the movie



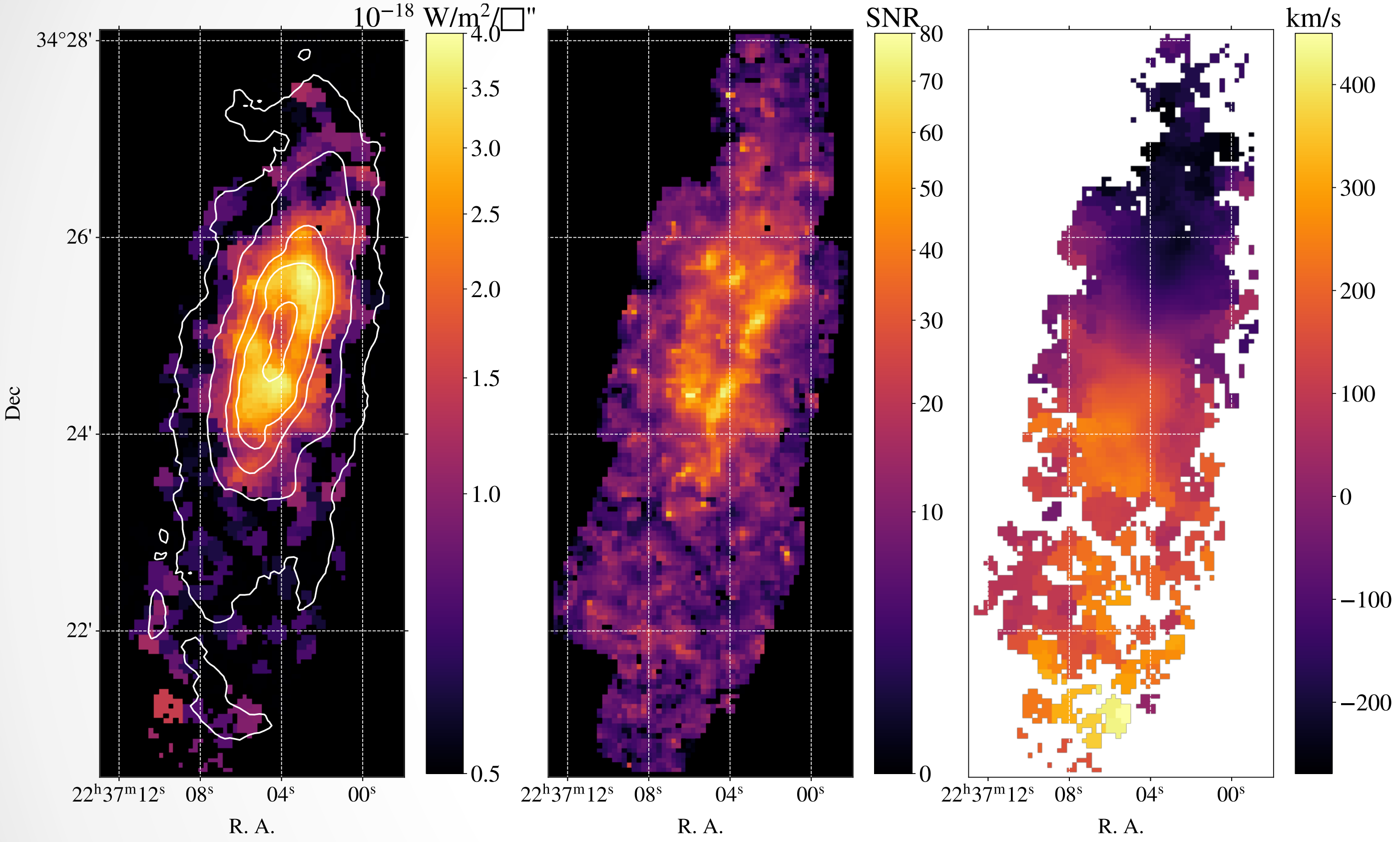
Dario Fadda
SOFIA data school - 16 Apr 2023

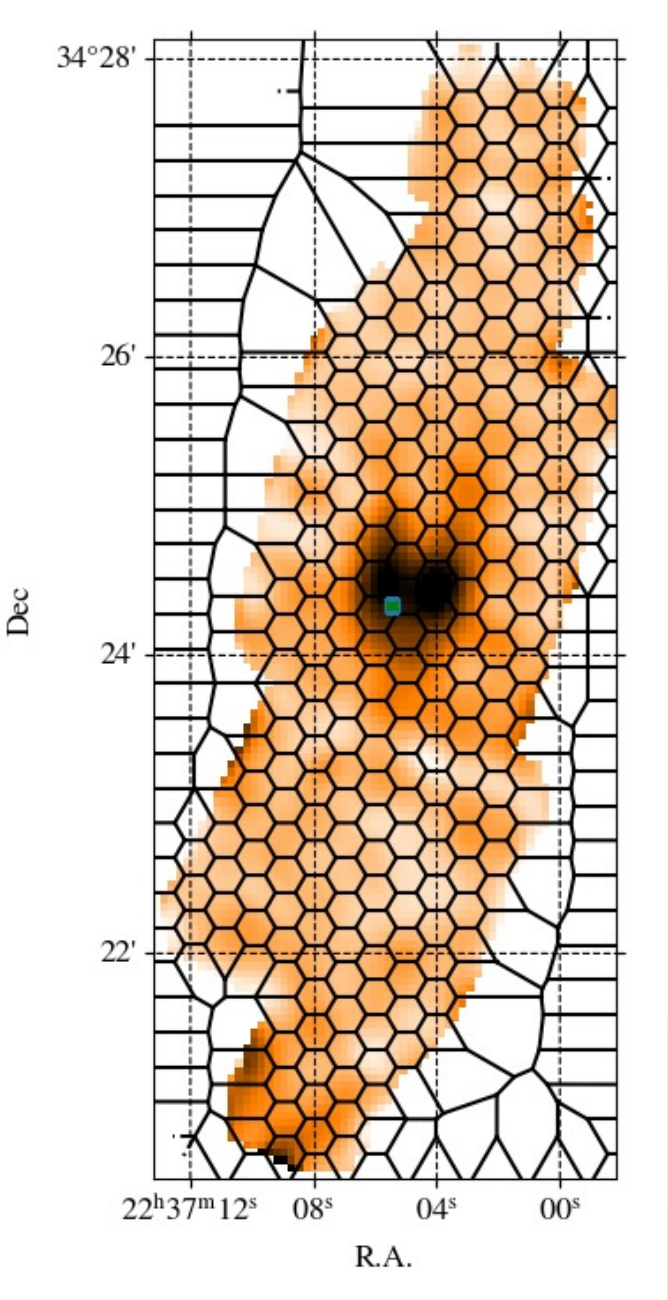
sospex: moments



Dario Fadda
SOFIA data school - 16 Apr 2023

Integrated flux





Better maps are obtained by specifying the initial conditions for the line fitting in many cells, as in this example of NGC 7331.

More about sospex

sospex allows one to make even more things, such as analyze Herschel/PACS cubes and spectral cubes from other instruments (i.e. ALMA).

It is possible to directly compare two spectral cubes (using the velocities in the x-axis for direct comparison of two lines).

Please, contact me if you have troubles using it, you want to leave a comment/suggestion, or you find bugs by submitting a ticket on github: https://github.com/darioflute/sospex/issues

The recent version (1.0.2) has been released to use Python 3.10 and the new version of matplotlib (3.7) which introduced many modifications.