KOSMOS MULTI-SLIT REDUCTION TUTORIAL

Abdullah Korra
The College of Idaho, Physics, Caldwell ID, United States.

A Multi-slit Spectroscopy Data Reduction Guide written by undergraduates, for undergraduates,
or anyone else interested in performing Multi-slit reduction
for the ARC 3.5m Telescope equipped with KOSMOS
(Kitt Peak Ohio State Multi-Object Spectrograph) instrument.

July 31, 2025

Contents

1. PREFACE

The purpose of this document is to guide a KOSMOS user through a spectroscopy data reduction, specifically the **multi-slit** reduction process. It assumes that the user is familiar with Python coding but does NOT assume the user has reduced astronomical observations before – imagine an undergraduate student starting astronomical research for the first time. The codes used in this reduction are based on the Pyvista astronomical Python packages built for spectroscopy reduction by Jon Holtzman. For more information, see the resources (2). **Throughout this manual, I will use the word "user" to refer to anyone who uses the pipeline.**

2. RESOURCES

- KOSMOS wiki for more information about the instrument and its usage: https://www.apo.nmsu.edu/arc35m/Instruments/KOSMOS/userguide.html
- Pyvista GitHub: https://github.com/holtzmanjon/pyvista
- Github repository with example data and notebooks (CofI_2025): https://github.com/mwbest/CofI_2025.git

3. SOFTWARE SETUP

This reduction is meant to run on a Jupyter Notebook because of Pyvista's interactive image visualization. However, it can also run in Google Colab with a few modifications. If you are running a pipeline on the Ubuntu operating system, make sure you are doing the following steps in a Python virtual environment. For Windows, that does not matter.

- Install Python (version 3.12.3 or higher), and Jupyter notebook on your local computer.
- Install Git on your local computer. The instructions for this installation can be found here https://github.com/git-guides/install-git.
- The CofI Reduction package is designed to be installed into your Python environment. The package includes all necessary modules for the reduction workflow.
- Navigate to the root directory of the project (‘CofI_multi/‘) in your terminal.
- Run the following command to install the package and its dependencies, **”pip install .”** or **”pip install -e .”** for editable.
- **(Optional)**: To install pyvista (astro-pyvista), follow the link; it will tell you all you need to know about its installation https://pyvista.readthedocs.io/en/latest/installation.html. For quick installation, run the following on the command line:

```
# If the current astro-pyvista is giving the user an issue.
pip install astro-pyvista # 0.4.1 is the stable operating version
```

- Through the command line, clone CofI_2025 on your device (https://github.com/mwbest/CofI_2025.git).

```
git clone https://github.com/mwbest/CofI_2025.git
```

Our GitHub repository CofI_Abdullah_2024 includes:

- A notebook of multi-slit reduction of the KOSMOS. Steps: Each step is explained by this workflow. The notebook name is ('widget_notebook.ipynb'); a link to the notebook https://github.com/mwbest/CofI_2025/blob/main/widget_noteboook.ipynb
- A half night of KOSMOS data (in the 'Star_JH-21Y252' folder) and their associated input mask files (in the 'kms' folder), to be used by the workflow notebook and other tests; and
- a set of wavelength reference files in "new_wave_lamp" folder, optimized for the specific spectrograph configuration and set of lamp observations taken during the half night included above: other users may need to update these files, potentially using the extensive library of KOSMOS arc lamps and line lists curated by James Davenport at https://github.com/jradavenport/kosmos-arc.

Running the notebook

Upon cloning CofI_Abdullah_2024, the clone folder will have the sample data and the tutorial notebook described above. To open the notebook in Jupyter, open the directory CofI_Abdullah_2024 in a terminal and write "jupyter notebook." A window named Home should pop up in your default browser. In the home tab, you can either run a new notebook and do your reduction following the manual, or you can run the existing tutorial notebook named "workflow_draft.ipynb" for a trial run. If the user has JupyterLab APP running locally, then navigate to the folder and run the notebook.

4. MULTI-SLIT IMAGE

Before introducing what a multi-slit image looks like, let's define several key terms:

Astronomical image: An astronomical image is a visual representation of data collected from observations of celestial objects and phenomena. These images are two-dimensional arrays of values. With the telescope shutters opened, it gathers light through the camera pixels. Ideally, each pixel should represent an element of an array whose value corresponds precisely and exactly to the amount of light source detected from an astronomical object at that position in the sky. Unfortunately, that is not the case due to the telescope's fundamental preset bias characteristics and some celestial interference. Thus, we have to reduce the images to remove these effects and accurately measure the light received from the celestial object being observed.

FITS Files: Astronomical images are often stored in Flexible Image Transport System (FITS) files, which include both the image data and metadata about the observation, such as the time, location, date, instrument settings, and calibration information.

Spectroscopy: a measurement technique based on the principle that different substances absorb and emit light at specific wavelengths. By examining these wavelengths, astronomers can identify and quantify the elements and molecules present in a sample.

Spectroscopy in Astronomy: Spectroscopy is crucial in astronomy for analyzing celestial objects' composition, temperature, density, and motion. It helps in studying stars, galaxies, and interstellar matter.

Single Slit and Multi-slit Spectroscopy: At the Apache Point Observatory, the ARC 3.5 meter telescope uses the KOSMOS (Kitt Peak Ohio State Multi-Object Spectrograph) instrument to collect light from objects projected onto one or more slits. That light gets dispersed through a grism disperser to spread the source's light out as a function of wavelength. **Single slit** observations are simpler to acquire and reduce but only allow the observation of one target at a time. Alternatively, one can observe with a 'slit mask' containing multiple slits, placed carefully to enable the collection of spectra from multiple objects at a time. These masks must be aligned carefully during observations to ensure all the stars fall on the intended slits, but once they do, we can collect data for multiple stars at once. Thus, **multislit** reduction (see Figure 1).

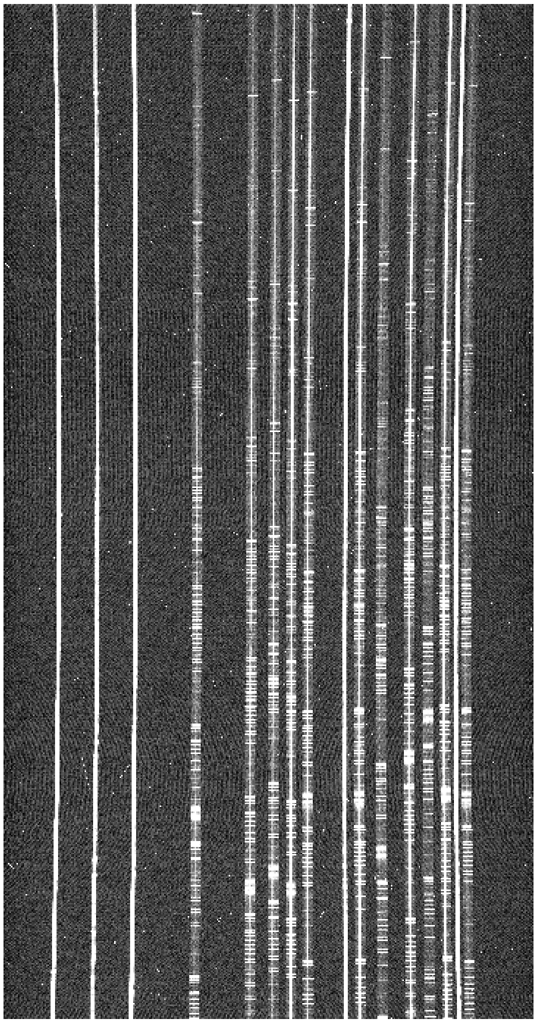


Figure 1: This is what a raw multi-slit image looks like for KOSMOS, with the positional axis oriented horizontally and the wavelength axis oriented vertically.

5. CALIBRATION IMAGES

!Note: If the user is familiar with spectroscopy data reduction processes, then skip the interface section for details on how the interface works. The following sections explain the basic concepts of spectroscopy data reduction procedures.

Astronomers use calibration images to correct for things like readout noise (offset noise), thermal noise, electric noise, and uneven brightness in science images. Calibration images are, **Bias, Dark Flat, arc lamps**.

5.1. *Bias*

Bias is an offset voltage value added to the pixels of a telescope camera to ensure that when images are taken, they do not have a negative value. At the same time, the initial brightness of the pixels affects science images; therefore, they need to be subtracted. A bias image (see figure 2) is taken in the shortest time the telescope (instrument) can with the shutter closed. The shortest possible time for the KOSMOS instrument is zero seconds.

To subtract any calibration image from our raw science images, we need a master or super bias, darks, and flats. We also take into account something called overscan. The Overscan region is a part of the camera's detector that is covered and receives no light from an observing source. However, the chip(s) will register as if they detected a certain amount of light due to the offset value. Thus, the level of this offset must be removed from all calibration and science images. Overscan removal happens in the background in this pipeline.

The code below will only remove the overscan from the bias image and display the image (see Figure 2).

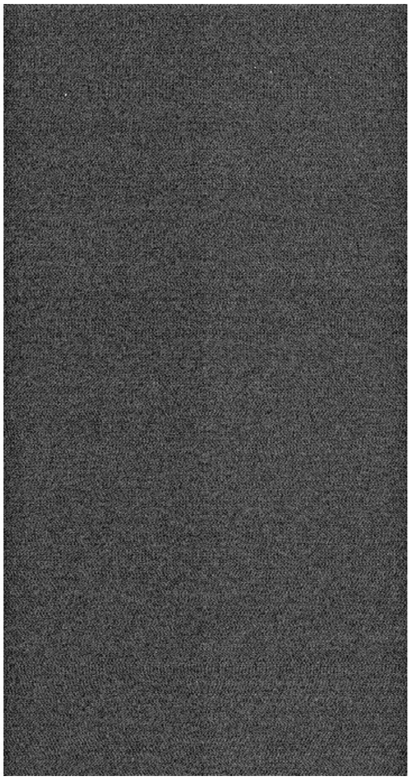


Figure 2: A single bias image.

5.1.1. *Master bias*

A master bias (super bias) combines all the biases to reduce the noise in the measured bias level. This ensures that as bias gets subtracted from the science image, they are not subtracting anything else except the bias level.

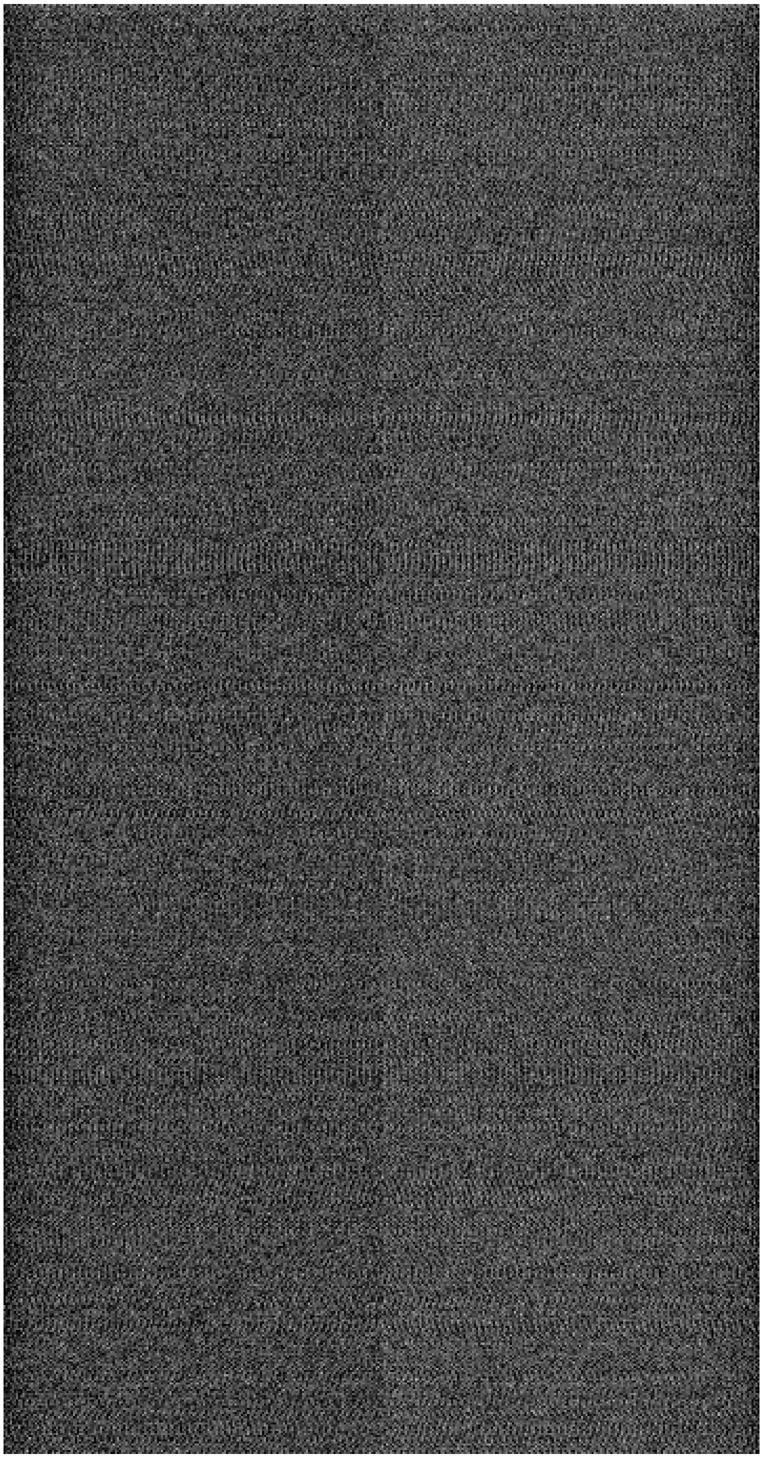
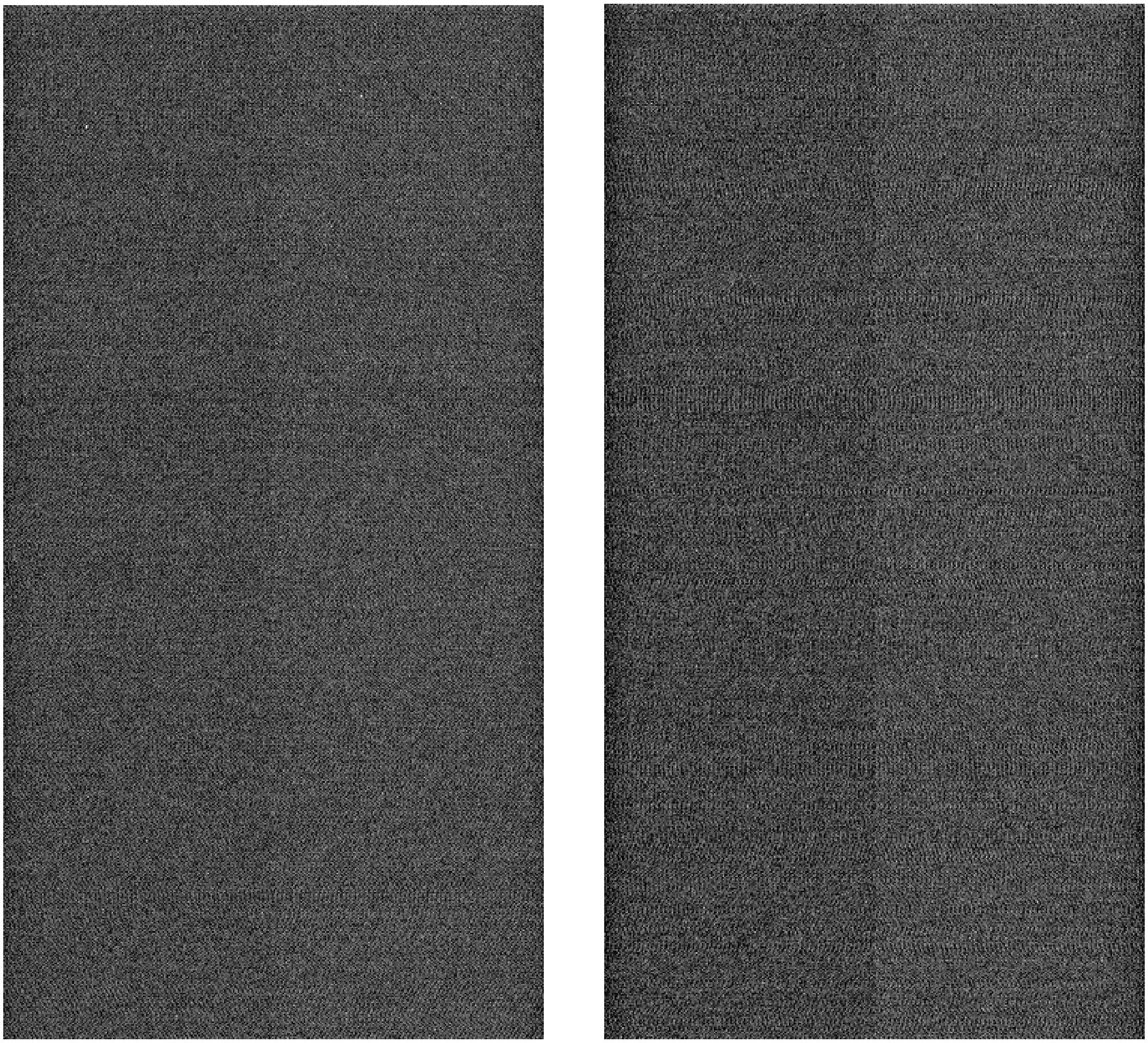


Figure 3: Master bias

The images below show a visual difference between single bias and master bias. Inspecting them closely, one can notice a few differences between them.



(a) Single bias.

(b) Master bias

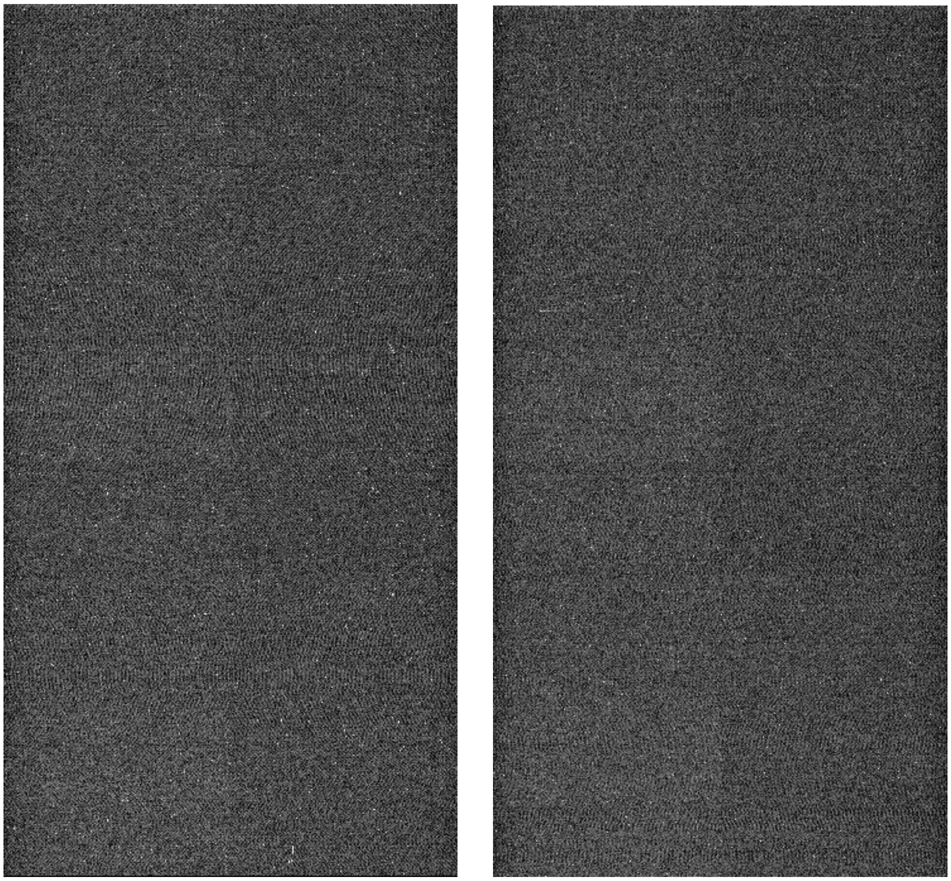
Figure 4: This is a comparison between a single bias frame (left) and a master bias frame constructed from multiple exposures (right).

Important note Bias:

- KOSMOS does have pattern noise, and it can change from bias to bias, so it isn't repeatable enough to produce a reliable and robust bias subtraction.
- We are currently averaging over 11 biases, which averages out most of the pattern noise, so it probably isn't ADDING much noise, but it probably isn't helping either.

6. DARK

Darks correct for thermal noise, hot pixels, and electric noise. As we take science images, as time goes on, the pixels begin to get hotter and brighter. Thus, the concept of thermal noise and hot pixels. An example of electric noise will be some electrons moving at high velocity compared to the rest, causing a mess in the pixels. All of these are registered in science images as they are taken. Therefore, they need to be removed. A dark can be taken before or after your science image using the same exposure time as the science frame but with the shutter closed. The process of making a master dark is no different from that of master bias.



(a) A single dark.

(b) Master dark.

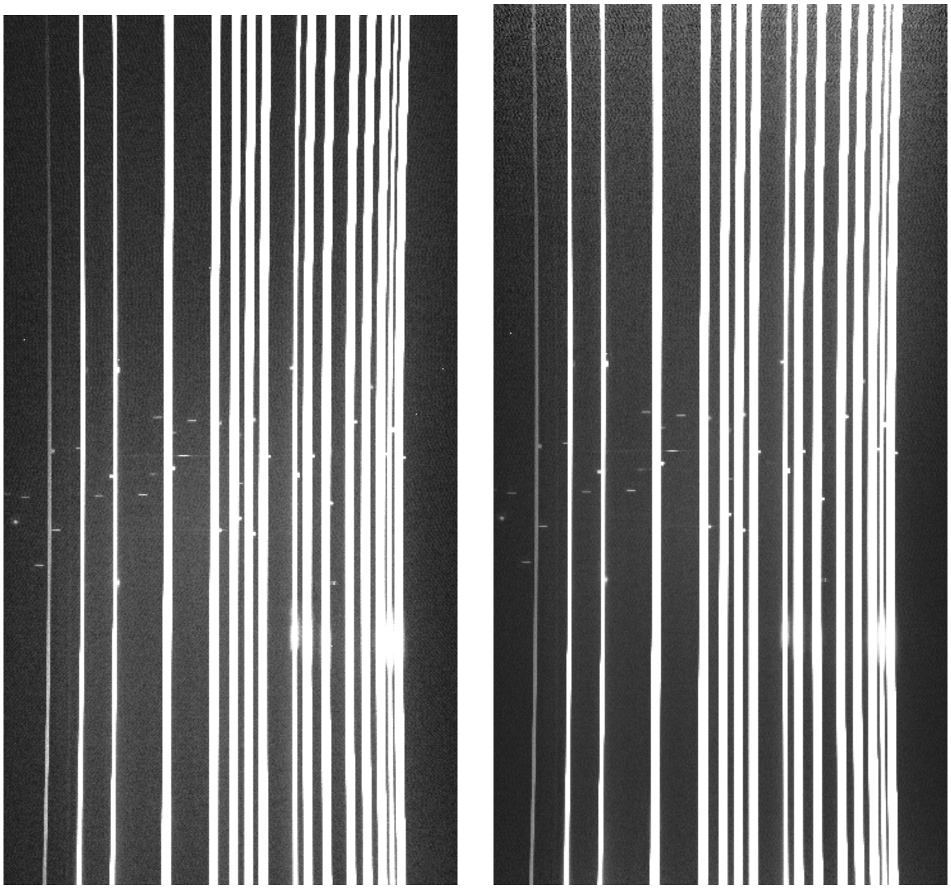
Figure 5: A visual comparison between a single dark and master dark.

Important note on Darks:

- Dark subtraction - can be useful, but it is still hard to get enough darks to reduce the readout noise in the combined dark frame (we were currently averaging 3 x 20m exposures, so we are only beating readout noise down by root(3))
- The recommendation is to use the ’clip’ method in ”mkdark()” function to only apply the correction to pixels that are 5 sigma (or whatever is set from the clip parameter) detections of significant positive dark current.

7. FLAT

When one takes a science image with the shutter open, dust or marks of some sort will be on some pixels. These marks reveal that dust or other optical/environmental effects are absorbing some of the light before it gets to the detector. Flats are taken to measure the shape and amount of light that are lost in these marks so that the lost light can be accounted for and corrected in science images. Flats are taken with bright white light spread over a white surface, which would ensure the same brightness across the pixels in the absence of these dust marks or optical effects. Science images can then be divided by the flat to correct for the uneven illumination that these dust particles and optical effects produce in the pixels in science images.



(a) Single flat

(b) Master flat.

Figure 6: A visual comparison of a single flat and a master flat

Note: The master flat process is the same as the master bias and dark but simultaneously different. If desired, one can subtract a master bias and master dark from a flat. But the combining process for a master flat is the same.

Important note on Flats:

- The recommendation is taking flats through a slit mask: since detector response is wavelength dependent, it's important to actually get the right wavelength as projected through the slit mask (which will be different than if we used the long-slit flat).

- cosmetic issues due to the motion of the slit are probably not a big deal (but the reduction could inch into the slitless to try and avoid the division issues at the edges).
- Littrow ghost issues are... complicated. Jon might think about this, but the best mitigation at the moment is to visually look for the artifacts in the flat and then be cognizant that the final spectrum might have been over divided by the excess of flux flat, so it will have a deficit of flux in the target spectra.

8. ARCS

Arcs refers to the calibration lamps. Calibration lamps are light sources that are used in spectroscopy to ensure the accuracy and reliability of measurements. They are essential tools for calibrating the wavelength scale of spectroscopic instruments. There are several types of calibration lamps, each serving different purposes. In our case, we are using neon lamps. Make a master arc using the code below. The master arc is made using the same combiner as master bias and master dark, but instead of median, it uses sum.

Looking at the master arc (see figure 7), you will notice that each slit has its portion of light. This means each object must be calibrated with its corresponding arc spectrum.

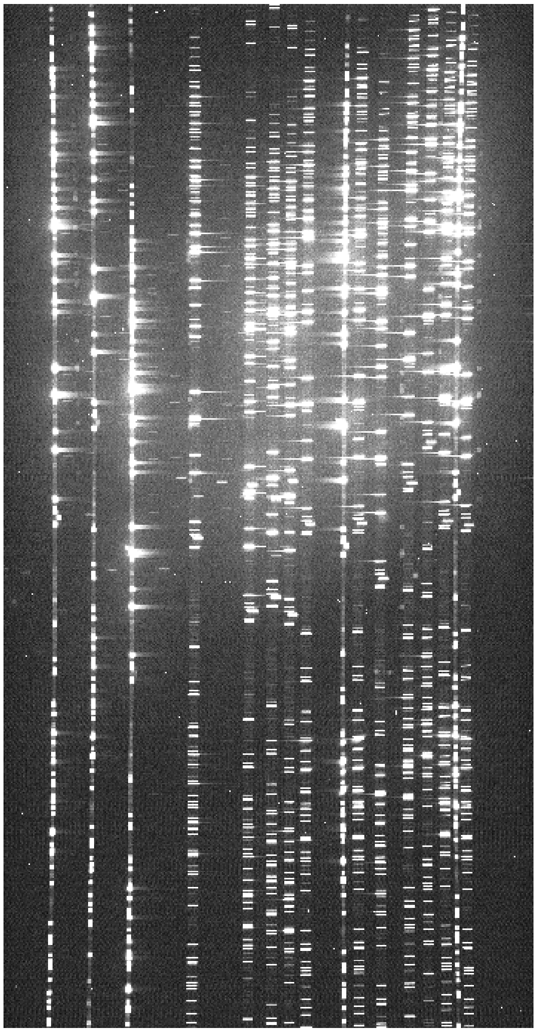


Figure 7: Master arcs

9. INTERFACE USAGE

This guide explains the functionality of the user interface, with a particular focus on our selected default settings. The interface is designed to make data reduction intuitive and straightforward, minimizing the need for user coding. The UI is structured into six main tabs. All tabs, except for ”User Guide” and ”Data Input,” are divided into subtabs dedicated to specific computational tasks. The overall workflow is organized into three primary sections: reduction, spectra stacking, and radial velocity calculation.

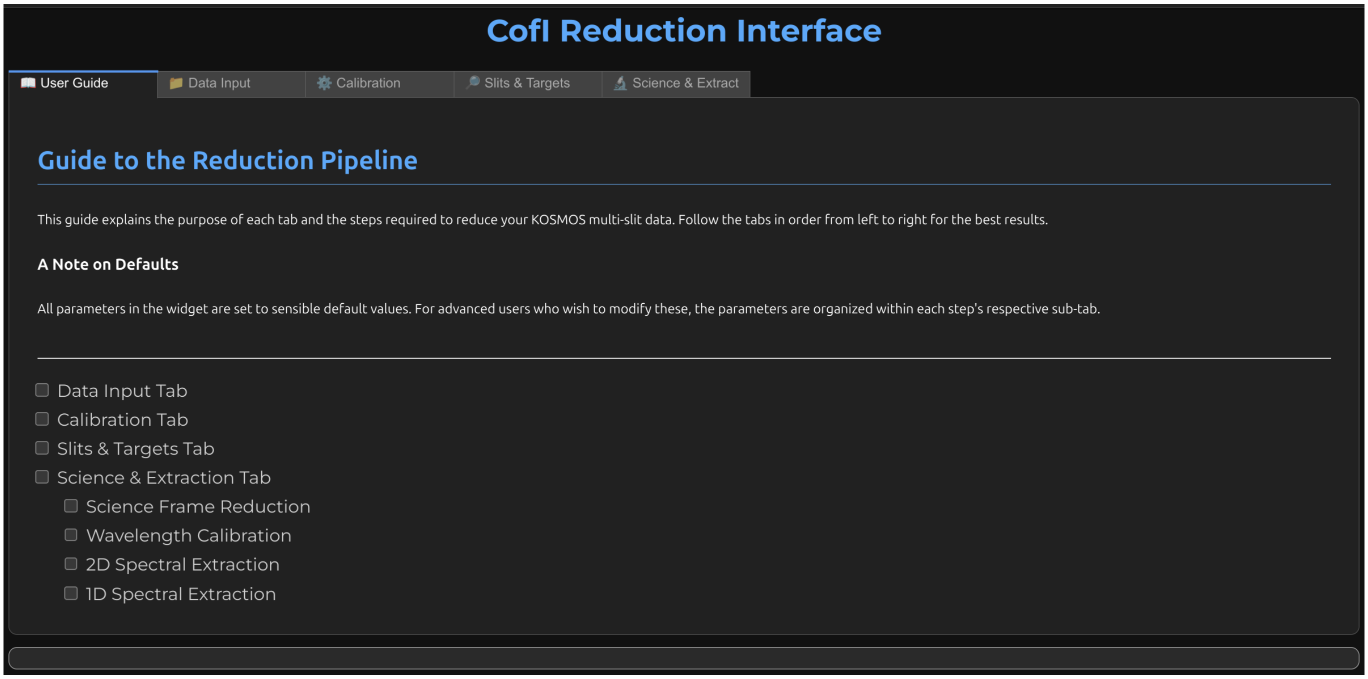
For a comprehensive review of all parameters, please consult the official astro-pyvita documentation by Jon Holtzman: https://pyvista.readthedocs.io/en/latest/.

9.1. *Launching the Widget*

Running the code below launches the User Interface (UI) in a separate window outside of the notebook (Figure 8). This new window is interactive, allowing you to work directly with your images as seen in Figure 9. From this point on, all plots generated will appear in a pop-up window. Run the import cell first before launching the UI.

Run the cell below

```python
reduction_widget = CofiReductionWidget(display_enabled=True)
# Display the widget
reduction_widget.show()
```



Guide to the Reduction Pipeline

This guide explains the purpose of each tab and the steps required to reduce your KOSMOS multi-slit data. Follow the tabs in order from left to right for the best results.

A Note on Defaults

All parameters in the widget are set to sensible default values. For advanced users who wish to modify these, the parameters are organized within each step's respective sub-tab.

- □ Data Input Tab
- □ Calibration Tab
- □ Slits & Targets Tab
- □ Science & Extraction Tab
  - □ Science Frame Reduction
  - □ Wavelength Calibration
  - □ 2D Spectral Extraction
  - □ 1D Spectral Extraction

Figure 8: Guide Tab; it acts as a built-in quick-start manual. Use it to find a summary of what each tab does, helping you quickly understand their functions or remember specific steps in the workflow.

Pop-up of the main display window

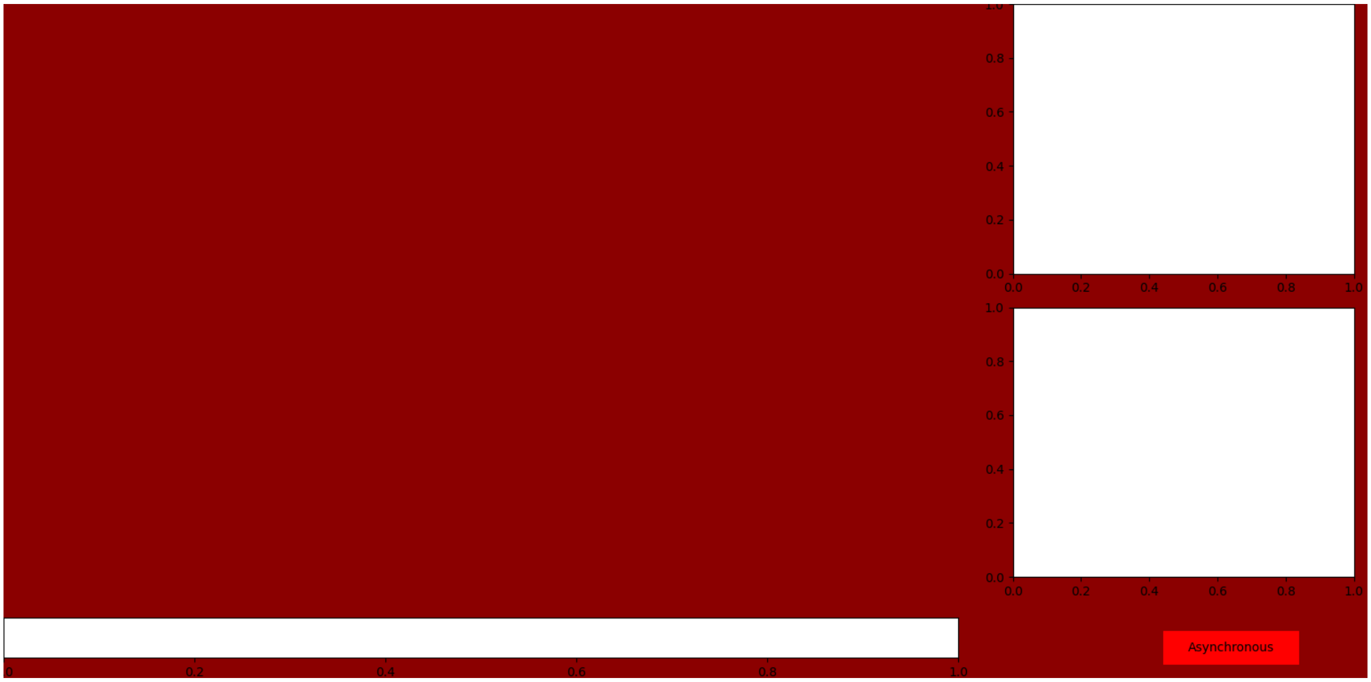


Figure 9: The main display window should not be closed, but adjacent windows can be. If the main window is closed, cells requiring it will produce an error. To fix this, re-run the launching cell.

9.2. *Data Input Tab*

The widget is organized into a series of tabs that follow the logical flow of a multi-slit data reduction. It is recommended to proceed through the tabs from left to right.

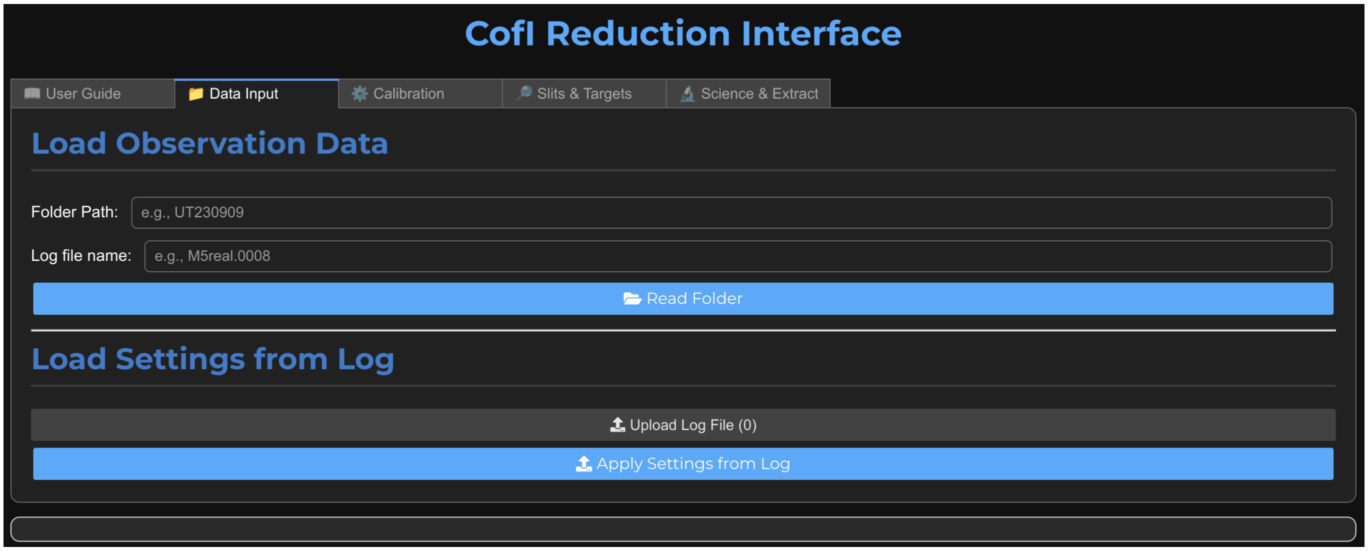
This is the first and most crucial step, where you direct the pipeline to your raw data files (see Figure 10).

- **Folder Path:** Enter the directory path that contains your raw FITS files (e.g., bias, dark, flat, arc, and science frames).
- **Log file name:** Specify a base name for the log file that will record all parameters used during your session. This is essential for reproducibility. The same name will be used to create folders for storing the 2D and 1D extractions sciences.
- If the name M3_7 is provided, the code will do the following:
  - **Log file name:** M3_7_log_2025-07-28.txt
  - **Extracted 2D spectrum folder name:** M3_7_2d_extractions
  - **Extracted 1D spectrum folder name:** M3_7_1d_extractions
- If no name is provided, the code will do the following:
  - **Log file name:** _7_log_2025-07-28.txt
  - **Extracted 2D spectrum folder name:** _7_2d_extractions
  - **Extracted 1D spectrum folder name:** _7_1d_extractions

- **Read Folder:** Clicking this button initializes the reduction engine. It scans the specified directory and displays a detailed log of all FITS files found in the main output area below the widget.
- **Upload Log File (Optional):** You can upload a ‘.txt‘ log file from a previous reduction session.
- **Apply Settings from Log (Optional):** After uploading a log file, this button will automatically populate all fields in the widget with the parameters from that session, saving time and ensuring consistency.

Important Notes:

- **Log File Limitation**: Please be aware that when a log file is used, the extraction radius and background parameters will not be updated.
- **Next Steps**: If you are ever unsure how to proceed, when the widget is in the middle of computing an action, check the output area of the interface. It will provide instructions and guidance for the next step.
- **Recommendation**: Before proceeding, please check your project folder to ensure that all files are numbered properly and that there are no numbering conflicts. Taking the time to organize your files correctly now will prevent errors and save you a significant amount of time later.



Folder Path: e.g., UT230909

Log file name: e.g., M5real.0008

Figure 10: Data input tab

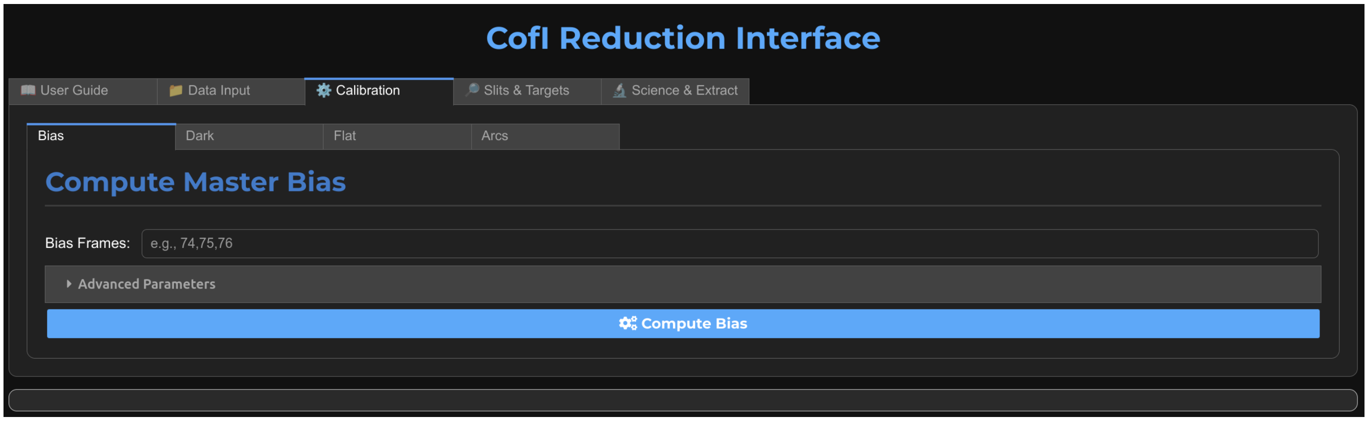
9.3. *Calibration Tab*

This tab is dedicated to creating the master calibration frames required to correct for instrumental signatures in your science data. The process is similar for each sub-tab (Bias, Dark, Flat, Arcs), Figure 11.

9.3.1. *Master Bias*

Workflow: Enter a comma-separated list of your raw bias frame numbers (e.g., 11,12,13,14,15,16) into the ”Bias Frames” field; This is the reason we asked the user to number the file properly because an alternative is entering the actual path of the files (e.g., bias . 0011. fits, bias.0012.fits, bias . 0013. fits, bias . 0014.

fits, etc). In the ”Advanced Parameters” section, you can select the combine method (e.g., median, mean). Click ”Compute Bias” to generate the master bias frame.



Cofl Reduction Interface

Compute Master Bias

Bias Frames: e.g., 74,75,76

Figure 11: Making master images of the calibration files

9.4. *Master Dark*

Workflow: Enter the file numbers for your dark frames. You can choose to subtract the master bias frame before combining. Advanced options allow you to clip data based on uncertainty. Click ”Compute Dark” to generate the master dark. **Recommendation**: Apply bias to the dark.

9.4.1. *Master Flat*

Workflow: Enter your flat frame numbers. It is recommended to apply bias and dark corrections. The ”Spectral Flat” option should typically be checked to normalize the wavelength-dependent shape of the lamp. Click ”Compute Flat”.

9.4.2. *Masftr Arc*

Workflow: Enter the file numbers of your arc lamp exposures (e.g., ‘23,24‘) and click ”Compute Arcs”. The code uses a sum combine method to create the master arc. No extra step needed.

Note: We recommended that users use calibration images wherever possible, because we are trying to reduce the noise as much as possible due to the low resolution of the camera. Going off by one pixel might cause the user uncertainty on 35 km/s. Thus, please apply your calibrations.

9.5. *Slits & Targets Tab*

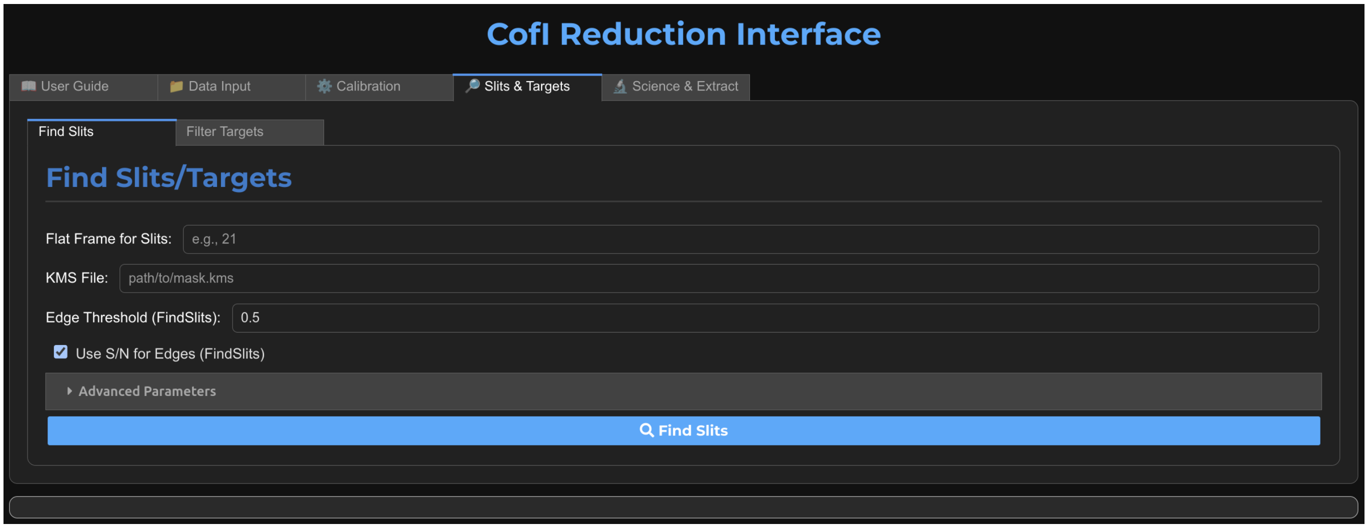
This tab handles the geometry of your observation: locating the slits on the detector and linking them to the intended targets from your mask design.

9.5.1. *Find Slits and Targets*

- **Purpose**: To identify the physical location of the slits on the CCD and correlate them with the target information from a KOSMOS slit mask (‘.kms‘) file (see Figure 12).
- **Workflow:**
  - **Flat Frame for Slits:** Enter the file number of a single, well-exposed flat where the slit edges are clearly illuminated.

  - **KMS File:** Provide the full path to the '.kms' file that corresponds to your observation.
  - **Find Slits:** This button uses the flat frame to trace the edges of each slit. It then reads the '.kms' file and displays a table of all targets, matching them to the found slits based on their vertical position ('YMM').

- **Importante Note**
  - If the code is unable to identify any slits, try increasing the Edge Threshold value. The default value of 0.5 assumes a high-quality flat-field image.
  - When some, but not all, slits are found correctly, it will cause an error during the 2D/1D extraction. This can happen when using a lower-quality flat. To resolve this, you must manually edit your KMS file to remove the entries for the slits that were not identified.
  - The best way to spot this is by paying attention to the output. The user should see warnings of the following:
    - **Warning:** Found bottom = 11 and found top = 11 the slits are not properly found
    - **Warning:** Found 11 slits but KMS file has 10 targets.



Flat Frame for Slits: e.g., 21

KMS File: path/to/mask.kms

Edge Threshold (FindSlits): 0.5

☑ Use S/N for Edges (FindSlits)

Figure 12: Finding slits

9.5.2. *Filter Slits & Update Headers*

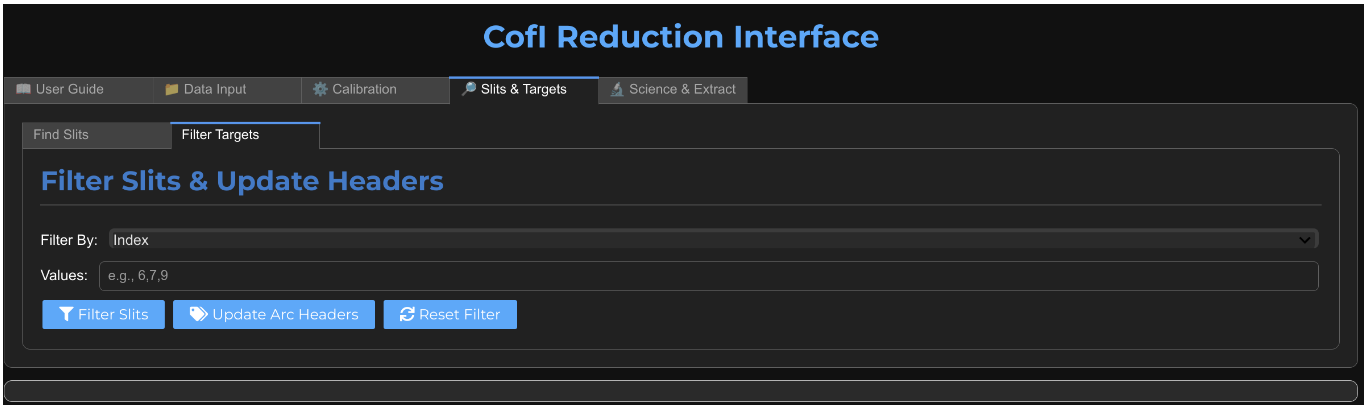
- **Purpose:** To select a subset of targets for reduction and to embed necessary coordinate information into the arc frame headers for wavelength calibration.

- **Workflow:**
  - **Filter Slits (Optional)**: If you do not wish to reduce all targets, you can select a subset. Choose a method (Index, ID, or Name), enter a comma-separated list of values, and click "Filter Slits". Use the "Reset Filter" button to revert to the full list.

    - **Filtering process:**
      If some slits lack a light source or you only need part of the science images, identify and pick the index of the object of interest relative to others in the image. Slit numbers are counted from left to right and match the table above. Remember, Python uses zero-based indexing. In the cell below, you can remove unneeded slits without affecting the original kms-file. The function allows filtering the KMS-file by indexes, target IDs, or names.

    - **Targets selection process:**
      - First, select the method of interest from the dropdown.
      - Then insert your stars of interest.

      The cell returns the trace and targets of the selected stars. You can use the example indexes and target IDs provided. Names are not included here since our stars lack names, but you can filter by names if the user's stars are named.
    - **Samples to use for trial run using the sample data:**
      - indexes: 11,8,7,9
      - ID: TARG101,TARG102,TARG106,TARG107
  - **Update Arc Headers**: This is a critical step. Clicking this button extracts a 2D arc spectrum for each selected slit and writes its corresponding mask coordinates (XMM/YMM) into its FITS header. This information provides the initial guess for the wavelength solution.
  - **Reset Filter**: Allows user to try again in case they made a mistake during filtering.


CofI Reduction Interface

Filter Slits & Update Headers

Filter By: Index

Values: e.g., 6,7,9

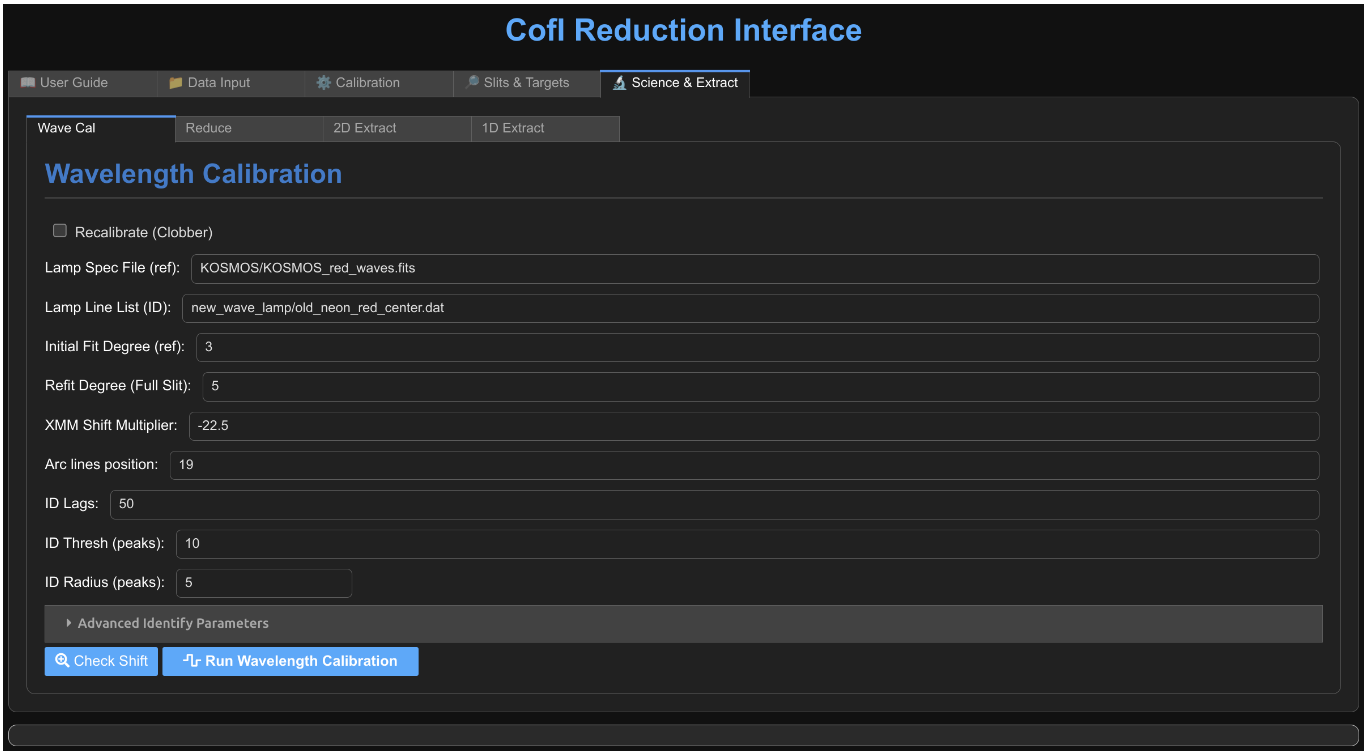
Figure 13: Filtering slits

9.6. *Wavelength Calibration*

Purpose: To determine the precise mathematical relationship between pixel position and physical wavelength for each slit. The wavelength calibration is a one-time thing that the user does not have to do over and over. Even though users have multiple solutions for multiple masks, the code is smart enough to know which calibration is used with which mask (see Figure 14). This is one of the most important steps in that reduction; thus, read through this section attentively.

Workflow:

- **Check Shift (Optional)**: This tool plots your extracted arc spectrum against a reference lamp spectrum. It provides a visual check that the initial guess for the wavelength shift is reasonable.
- **Run Wavelength Calibration**: This is an interactive step that uses the 2D arc spectra to identify known emission lines and fit a polynomial solution. For each slit, an interactive plot will appear where you can remove outlier points to improve the fit. A FITS file containing the solution for each slit is saved to the project folder.
- **Recalibrate (Clobber)**: Check this if the user wishes to redo calibrations of the mask they are looking at.



Wavelength Calibration

☐ Recalibrate (Clobber)

Lamp Spec File (ref): KOSMOS/KOSMOS_red_waves.fits

Lamp Line List (ID): new_wave_lamp/old_neon_red_center.dat

Initial Fit Degree (ref): 3

Refit Degree (Full Slit): 5

XMM Shift Multiplier: -22.5

Arc lines position: 19

ID Lags: 50

ID Thresh (peaks): 10

ID Radius (peaks): 5

Figure 14: Sub-Tab; Wavelength Calibration

Wavelength shift concept explained by Jon Holtzman

”Now loop through each extracted arc to do wavelength calibration. This requires a little effort because the change in the location of the slit relative to the default saved wavelength calibration is significant enough that it can be a challenge to automatically find the lines since the change in spectrum is more than a simple shift (and, in fact, more than a shift + dispersion change).

However, a simple shift is usually enough to identify some of the lines, and these can be used to bootstrap the wavelength solution; the initial identification is easier if you use an estimate of the shift from the mask design XMM.

You can really help this process if you supply an initial wavelength calibration (a pyvista WaveCal object) that was done using the same lamp(s) as your arc exposures (here, using KOSMOS_red_waves.fits’), and using a master line list that corresponds to these lamp(s) (here, using ne.dat). If you choose another WaveCal to start from, you may need to get a correct approximate relation for the shift from the reference spectrum as a function of XMM. For KOSMOS, it seems like -22.5(XMM) gives a rough pixel shift from a center slit location, -22.5(XMM-24.44) for a low slit location.

The next cell just shows how the simple shift fails, even using the XMM for each slit to shift: all of the arc lines don't match up with just a translation."

To understand this better, try changing the -22.5 value and see what happens to the plot. If the shift is way off, this means that your arc lamp might not be that good. To get around this, increase the **lag** value to about 100 or so in the wavelength calibration step.

Wavelength calibration steps: As the user runs the previous cell, an interactive plot window should pop up with identified lines. The plot should have a third-degree polynomial curve shape (see figure 15). If not, click "I" until you get the shape or something close to it (if the arc is bad). If clicking the keys does nothing, stop the code and re-run it. After getting the shape, use the following method to remove points that do not lie on the curve's path.

Input in plot window:

- l : to remove all lines to the left of the cursor.
- r : to remove all lines to the right of the cursor.
- n : to remove line nearest cursor x position.
- i : return with True value (to allow iteration).
- anything else: finish and return.

Remove points till the RMS is below one (1). Then "I" to re-identify lines if the new identification looks like figure 16. Proceed to the next-arc spectrum; if not, repeat the previous steps.

Output 1

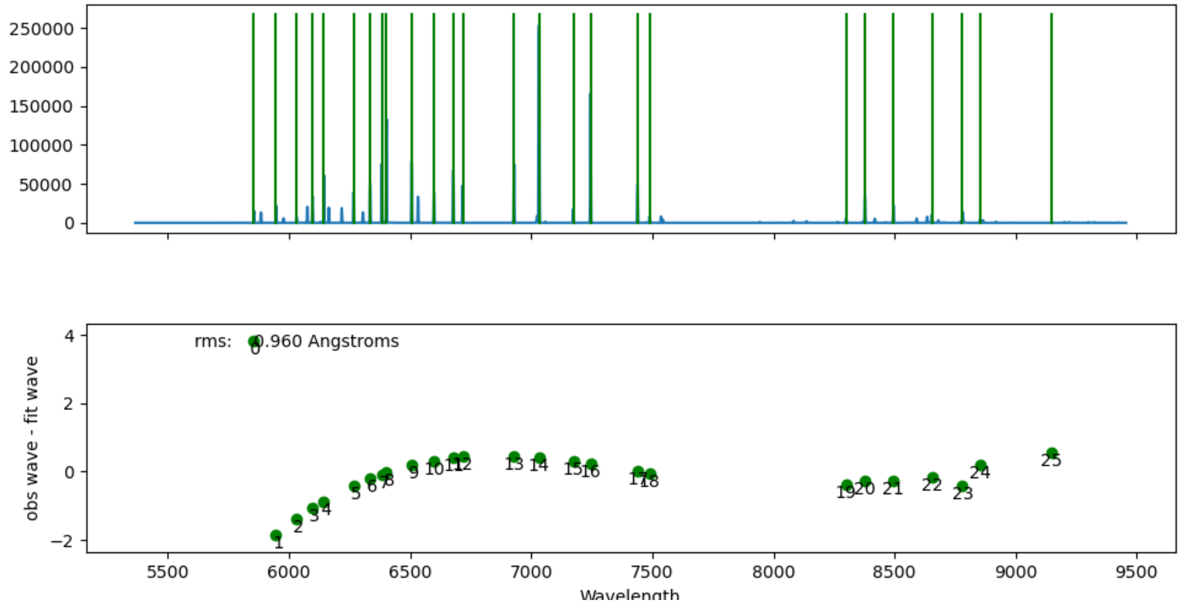


Figure 15: The polynomial curve should look similar or close to it.

Output 2

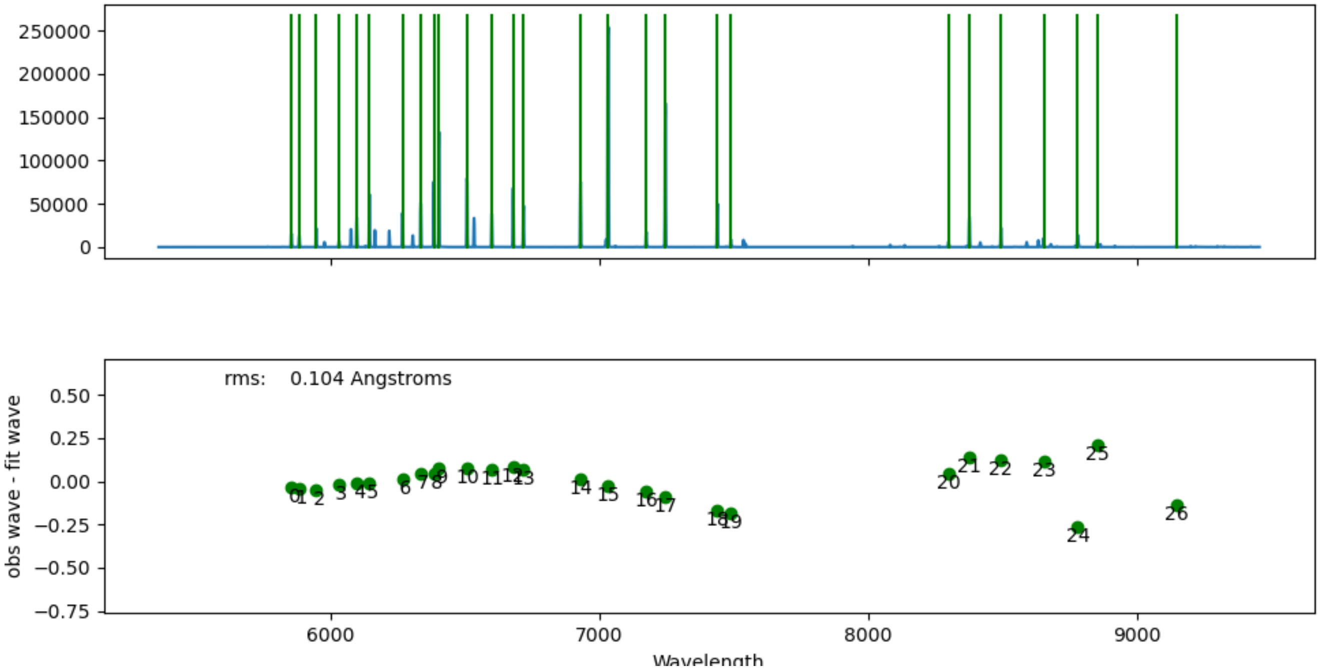


Figure 16: An example of a successful line identification plot. Although you can remove outliers here, like 24 and 26, it is recommended to leave them, as they will be handled automatically in the next step.

It will then take identified lines and sample them at ten places across the slit (see Figure 17)

Output 3

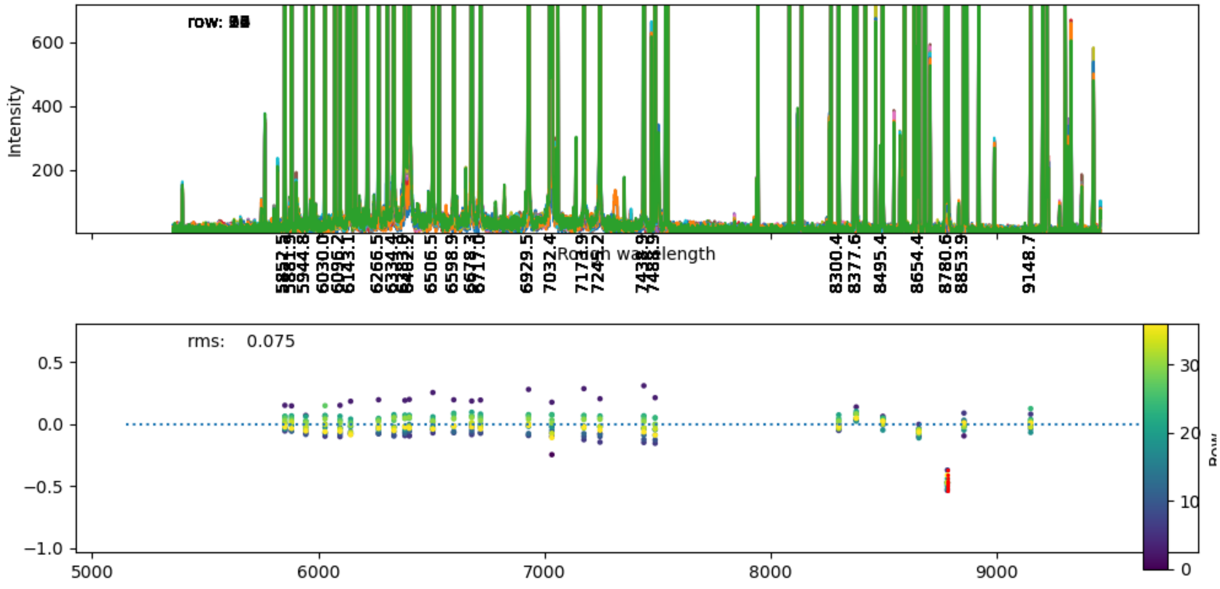


Figure 17: Sampling the identified lines at ten places across the slit for better wavelength solution.

9.7. *Science Extraction*

This is the core processing tab where the science frame is reduced and the final 1D spectra are extracted. The steps should be performed in order.

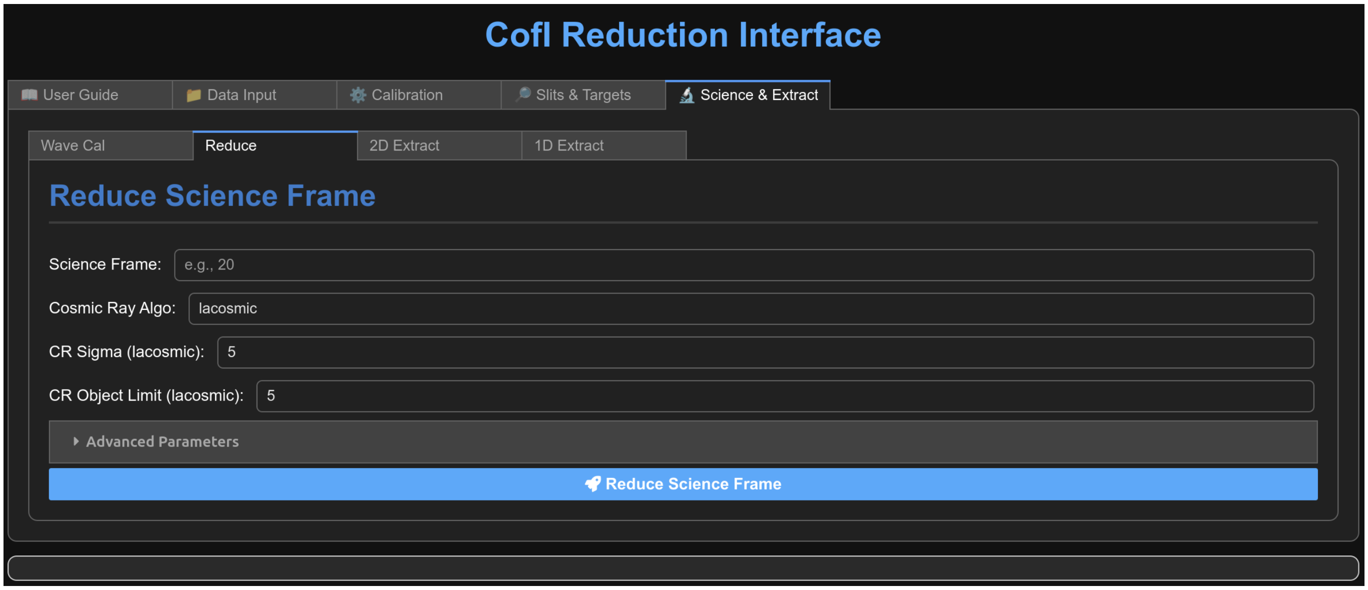
9.7.1. *Science Frame Reduction*

- **Purpose**: To apply all instrumental corrections to the raw science image. You can fine-tune cosmic ray rejection by adjusting the crsig and objlim parameters (see Figure 18). Always visually inspect the output image to verify the changes you've made.

- **Workflow**: This process calibrates the image and performs cosmic ray rejection to produce a clean 2D frame.

  - Enter the file number of your science exposure.
  - Check the boxes to apply the master bias and master dark frames created earlier.
  - Do not apply the flat frame at this stage.
  - **crsig (Laplacian-to-noise limit):** Lower this value to reject more cosmic rays.
  - **objlim (Contrast limit):** Increase this value to make rejection less sensitive. This is useful if you suspect that real data is being mistaken for cosmic rays.
  - **Best Practice:** For optimal results, the values for crsig and objlim should be kept relatively close to one another, but sometimes you can increase objlim a lot higher if necessary.

- **Applying Flats**: Applying the entire flat-field to the whole science image is not recommended because it will incorrectly brighten pixels both outside and inside of the spectral slits, potentially introducing noise and artifacts. Since the scientifically relevant data is contained only within the slits, the proper technique is to apply the correction on a slit-by-slit basis. This ensures each slit from the flat frame is applied to its corresponding slit on the science frame for an accurate result. Flats will be applied as the default during 2D extraction.



Science Frame: e.g., 20

Cosmic Ray Algo: lacosmic

CR Sigma (lacosmic): 5

CR Object Limit (lacosmic): 5

Figure 18: Sub-Tab; Science reduction

9.8. *2D Spectral Extraction*

Purpose: The goal of this step is to take the fully calibrated and cosmic-ray-cleaned 2D science image and extract the individual 2D spectrum (a ”slitlet”) for each targeted object. During this process, the wavelength solution derived from the arc lamps and the master flat from the flats are applied. You also have the option to perform a final wavelength adjustment using the night sky emission lines present in the science data itself.
Workflow:

The 2D extraction process is initiated from the ”2D Extract” sub-tab within the ”Science & Extraction” tab.

- **Set Advanced Parameters (Optional)**: Before starting, you can expand the ”Advanced Parameters” accordions to fine-tune the behavior of the underlying functions. These parameters control the tracing, peak finding, and skyline fitting algorithms. For most cases, the default values are sufficient. This is where we apply flats to science spectra. The option is under ”Extraction Parameters”.

- **Initiate Extraction:** Click the ”Setup & Run 2D Extraction” button.

- **Interactive Controls:** An interactive control panel will appear in the dedicated output area of the tab. This panel contains the primary choices for this step: ”No 2D wavelength adjustment” and 2D wavelength adjustment.

If ”No 2D wavelength adjustment” is selected (Recommended): This option will apply the existing wavelength solution from the arc frames without further refinement. The extracted 2D spectrum will be saved directly.

If ”2D wavelength adjustment” is selected: This is useful when the user has a bad lamp and can’t get the RMS value below one easily.

- **Identify Sky Rows:** Rows that are a certain distance (defined by the ”Skyline rad for rows” parameter) away from the object’s peak are designated as sky background.

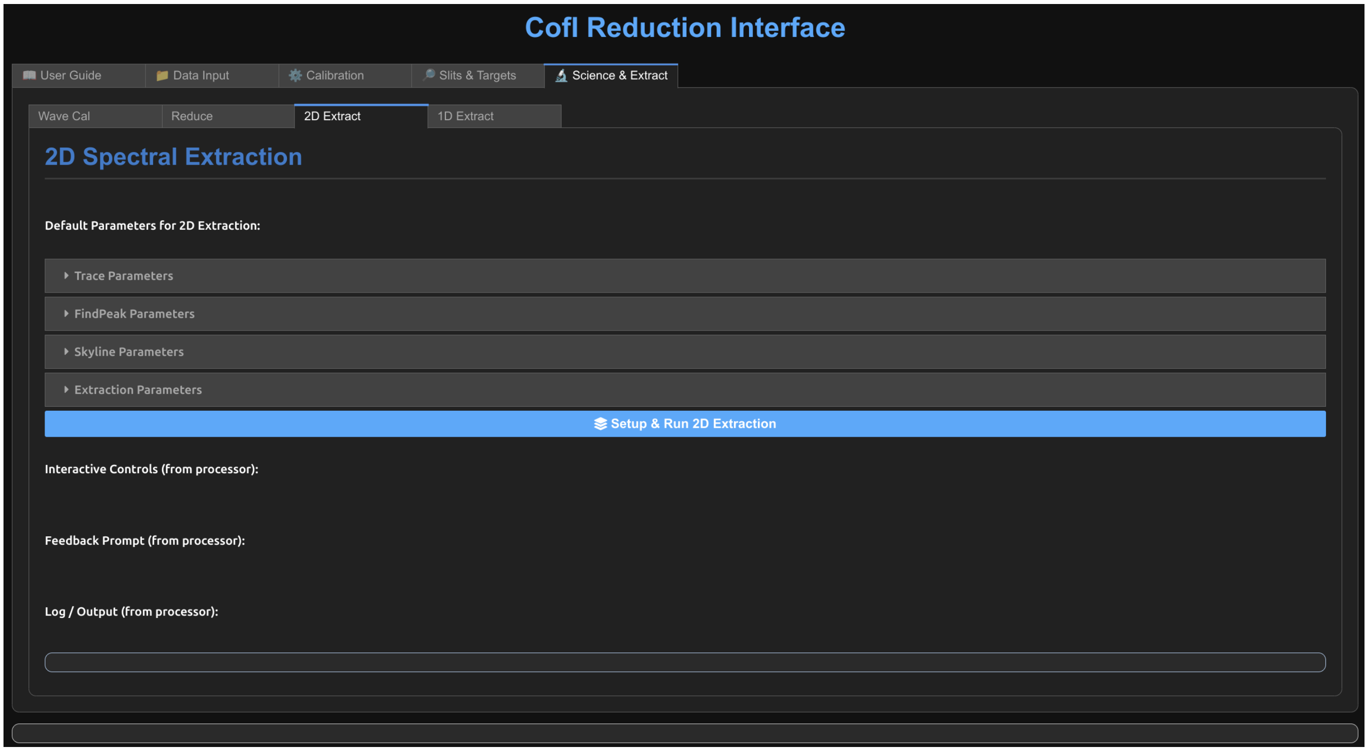


Figure 19: Sub-Tab; 2D extraction

- **Skyline Fitting:** The code analyzes these sky rows to identify well-known night sky emission lines (e.g., from Oxygen). It then calculates the offset between the observed positions of these lines and their known laboratory wavelengths. This offset is used to compute a final, subtle correction to the wavelength solution for that specific slitlet.

- **Interactive Prompt:** For each slit, an interactive plot appears showing the 2D spectrum with the identified science aperture and sky rows marked. You are prompted: ”Are you satisfied with the skyline result?”

  - **Yes:** The newly adjusted wavelength solution is applied, the 2D spectrum is rectified (making the spectral lines perfectly vertical), and the result is saved to a FITS file (e.g., ‘M3real_TARG107_2d.fits‘). The pipeline then proceeds to the next target.
  - **No:** You can select a new radius for the sky row selection and repeat the skyline fitting process for the current target.

Run Extraction: After making your selection, click the ”Extraction” button in the interactive panel to begin the process. The pipeline will then iterate through each selected target.

The Underlying Process:

- **Initial Extraction:** For each target, the code uses the previously defined slit traces to cut out the corresponding rectangular region from the reduced 2D science image.

- **Flat Fielding:** If a master flat frame was created, the corresponding slitlet from the flat is extracted and divided into the science slitlet to correct for pixel-to-pixel variations.

- **Applying Wavelength Solution:** The wavelength solution file (‘CofIwav_TARGET_ID.fits‘) saved during the ”Wave Cal” step is loaded and applied to the 2D slitlet, converting the xaxis from pixels to Angstroms.
- **Output:** The process generates a wavelength-calibrated, flatfielded, 2D FITS file for each target.

Important Note on ”Yes/No” Interactions
When you are prompted with a ”yes” or ”no” choice during 2D and 1D extraction, please follow this specific order if you are not satisfied with the result:

- **First**, make the desired changes to the parameters.
- **Then**, click the ”no” button.

The process will then re-run using your new settings. Clicking ”no” before making your changes will not work correctly.

9.9. *1D Spectral Extraction*

Purpose:
This is the final step, where the 2D slitlet is collapsed into a 1D spectrum representing flux versus wavelength. The process involves tracing the object’s spectrum, defining regions for the sky background, summing the object’s flux, and subtracting the sky.
Workflow: This process is initiated from the ”1D Extract” sub-tab (see Figure 20).

- **Set Advanced Parameters (Optional):** You can fine-tune parameters for tracing, peak finding, and skyline fitting in the advanced parameter accordions.
- **Initiate Extraction:** Click the ”Setup & Run 1D Extraction” button.
- **Interactive Controls:** A new control panel appears, allowing you to set the key parameters for this run:
  - **1D Calibration Choice:**
    - **No sky adjustment:** When the wavelength solutions are good, or you performed the 2D wavelength adjustment in the previous step, this is the correct choice.
    - **Sky adjustment:** If you chose ”No 2D wavelength adjustment” previously, and have bad wavelength solutions, you can use this option to perform a final wavelength calibration on the 1D spectrum using the sky lines. This is an old method, predating the development of the ”2D wavelength adjustment” method, which is based on the same concept.
  - **Extraction radius:** This defines the half-width (in pixels) of the aperture around the object’s trace from which flux will be summed.
  - **Bkg. Regions & Sky window offset:** These define the size and location of the windows on either side of the science aperture that will be used to calculate the sky background.
- **Run Extraction:** Click the ”Run Extraction” button to begin the automated process.
  - **Interactive Prompt:** For each target, a window appears showing the 2D spectrum with the trace and the extraction regions. You are prompted: ”Are you satisfied with the extraction?”

  - **Yes:** The 1D spectrum is finalized. If ”Sky adjustment” was selected, the 1D skyline calibration is performed. The final 1D spectrum and the measured sky spectrum are plotted, and the data are saved to a multi-extension FITS file. The pipeline then moves to the next target.
  - **No:** re-process the current target with the new setting.

- **Output:** The process generates a multi-extension FITS file for each target (e.g., ”1d_ad_5_M3real_TARG107_3.fits”). This file contains the final 1D science spectrum, its uncertainty, the wavelength solution, and the extracted sky spectrum.

Note: Due to the interactive nature of 2D and 1D extraction, there is a delay in updating radius, background region, background offset, and choice reduction in the log file. To ensure the integrity of reproducibility, we have made sure that updates are available in the final fits files’ header. The user can rest assured that the values in the fits header are always accurate because the code updates them in real time.

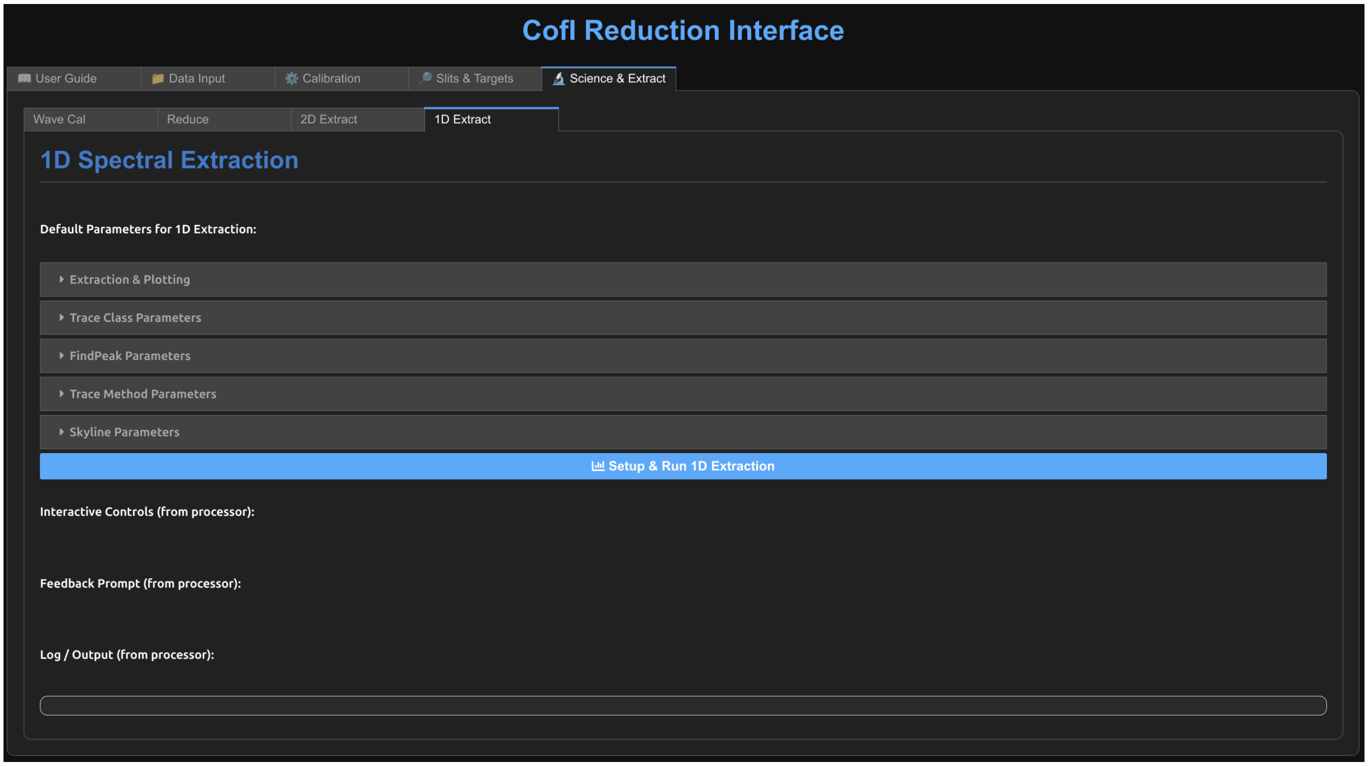


Figure 20: Sub-Tab; 1D extraction

A Note on Workflow Flexibility:

While it is recommended to follow the tabs sequentially from left to right when processing a new dataset, the true power of this widget-based interface lies in its flexibility—a significant advantage over traditional, code-heavy notebooks.

Unlike a linear, cell-by-cell script where modifying an early step often requires re-running everything that follows, this widget-driven approach allows for a more dynamic and efficient workflow. You can seamlessly switch between tabs to adjust parameters and re-run specific tasks without invalidating subsequent steps.

For example, if you are not satisfied with your initial 1D extraction, you can simply navigate back to the ”Calibration” tab, compute a new master flat with different parameters, and then return directly to the

”1D Extraction” sub-tab to process your science frame again with the updated calibration. This modularity encourages iterative refinement and experimentation, saving considerable time and effort.

However, it is crucial to remember that when beginning a reduction for the first time, especially with a new slit mask, adhering to the sequential order is the best practice. This ensures that all necessary data products (like master calibrations, slit traces, and wavelength solutions) are generated in the correct order before you begin refining the final science products.

10. POST-PROCESSING: SPECTRA STACKING

Purpose: If you have multiple exposures of the same science target, you can combine, or ”stack,” them to create a single spectrum with a higher signal-to-noise ratio (S/N). This process averages out random noise while preserving the real signal from the astronomical object, resulting in a cleaner and more reliable final spectrum. The module provides a simple interface to perform this task (see Figure 21).

Prerequisites: Before you begin, you must have at least two fully reduced, 1D extracted spectra of the same target. These should be the multi-extension FITS files produced by the 1D extraction step.

Launching the Stacking Tool

The stacking functionality is contained within the stacker function. To use it, import and call it from a new cell in your Jupyter Notebook:

```
#This will display the file input widgets
stacker()
```

This will render a simple user interface with two text boxes and a button (see Figure 21).

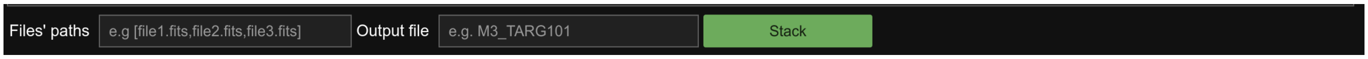


Figure 21: The UI for the stacking process

Workflow

- **Files’ paths:** In this text box, provide the full paths to the 1D extracted FITS files you wish to combine. The paths should be enclosed in square brackets and separated by commas, like a Python list of strings.
  - **Example:** [’Star_JH-21Y252/M3_0007_reduction/1d_ad_5_M3real_TARG107_3.fits’, ’Star_JH-21Y252/M3_0008_reduction/1d_ad_5_M3real_TARG107_3.fits’]
  - **Output file:** Provide a descriptive base name for the output file. The script will automatically prepend ”stacked_” and append ”.fits” to this name. The script will also create a folder that will contain the stacked spectra.
  - **Example:** M3_TARG107_stacked.fits
  - **Example:** If user used a Output file name M3_stacked the script will create a folder named stacked_M3
- **Stack Button:** Click this button to begin the stacking process.

Example **Important Note:** Do not stack spectra of poor quality (e.g., very low S/N) with high-quality spectra, as this can degrade the final result. It is often better to discard a poor-quality exposure rather than include it in the stack. Sometimes, even though all spectra are of good quality; but there are some that are of a lower quality compared to the rest; remove them.

11. POST-PROCESSING: RADIAL VELOCITY ANALYSIS

Purpose:

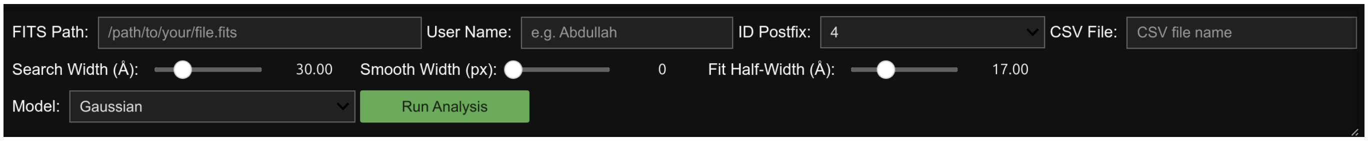
The AstroAnalysis class provides a comprehensive, GUI-based tool for measuring the radial velocity of a star from its 1D extracted spectrum. The analysis focuses on fitting the three prominent absorption lines of the Calcium II Triplet (CaT) at rest wavelengths of 8498.02 Å, 8542.09 Å, and 8662.14 Å. The tool automates the process of spectrum normalization, line fitting, and data logging, and then provides a secondary interface for calculating the final, combined radial velocity from multiple measurements.

textbfLaunching the Analysis Tool

The entire RV analysis workflow is encapsulated within the AstroAnalysis class. To begin, import and instantiate it in a new Jupyter Notebook cell.

```
#This will initialize and display the RV analysis widget
rv_analyzer = AstroAnalysis()
```

This will render the primary user interface (see Figure), which is used to analyze individual spectra and save the results.



FITS Path: /path/to/your/file.fits
User Name: e.g. Abdullah
ID Postfix: 4
CSV File: CSV file name
Search Width (Å): 30.00
Smooth Width (px): 0
Fit Half-Width (Å): 17.00
Model: Gaussian

Figure 22: The UI for the stacking process

The Analysis Workflow: A Step-by-Step Guide

The workflow is divided into two main stages: first, analyzing one or more spectra to gather line-fit data, and second, using that data to calculate the final radial velocity for a star.

Stage 1: Fitting the Ca II Triplet Lines

This stage is controlled by the main GUI panel. You will typically repeat this process for each 1D spectrum of your target star.

GUI Interaction and Workflow:

- Providing Inputs:
  - **FITS Path:** Enter the full path to a single, 1D extracted FITS file. This can be either a standard single-extension FITS file with a WCS header or a multi-extension file generated by the CofI reduction pipeline.
  - **User Name:** Enter your name for logging purposes. This will be saved with the results.
  - **CSV File Name:** Specify the name of the CSV file where the results of the line fits will be saved. If you leave this blank, a name will be automatically generated based on the object name

in the FITS header. All results will be appended to this file, allowing you to build a database of measurements. The script will also create a folder that will hold all the RV analysis results. E.g., if the User provided a name M3_stacked, it will create a folder named M3_rv_analysis.

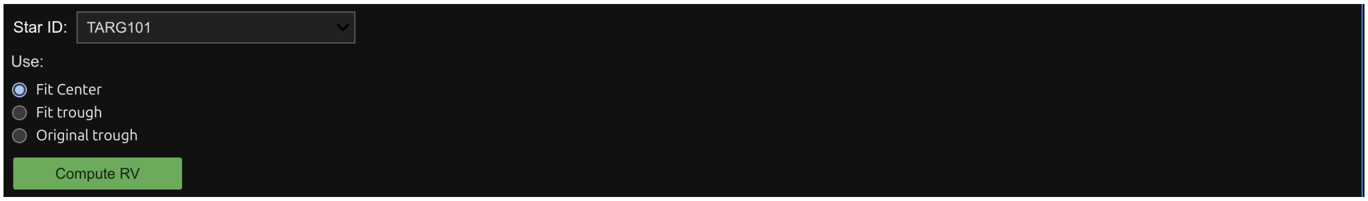
  - **ID Postfix:** This dropdown allows you to specify which part of the filename (split by underscores) corresponds to the target's unique ID. For a file like 1d_ad_5_M3real_TARG107_3.fits, a postfix of 4 would correctly identify TARG107 as the Star ID.

- **Setting Analysis Parameters:**
  - **Search Width (Å):** This slider defines the half-width of the wavelength window around each of the three CaT rest wavelengths where the code will search for the absorption line's minimum.
  - **Smooth Width (px):** You can apply a boxcar smooth to the spectrum before fitting. This slider controls the width of the smoothing kernel in pixels. A value of 0 means no smoothing is applied.
  - **Fit Half-Width (Å):** This defines the half-width of the smaller window around the detected line center that will be used for the actual profile fitting. This allows you to isolate the line core and ignore surrounding continuum features.
  - **Model:** Choose the profile to fit to the absorption lines:
    - **Gaussian:** A standard Gaussian profile.
    - **Voigt:** A more complex profile that is a convolution of a Gaussian and a Lorentzian profile, often better for describing stellar absorption lines.
- **Running the Analysis:** Click the "Run Analysis" button.

Stage 2: Calculating the Final Radial Velocity

After you have analyzed all your spectra for a given star and populated the CSV file with measurements, you can proceed to the final RV calculation. This stage is automatically handled by the calculate_radial_velocity method, which presents a new, secondary GUI (see Figure 23).

```python
#This will initialize and display the RV calculation widget
analyzer.calculate_radial_velocity('stacked_rv_analysis/stacked_M3.csv')
```



Star ID: TARG101

Use:

● Fit Center

○ Fit trough

○ Original trough

Figure 23: The UI for RV calculations

UI Interaction and Workflow:

- **Star ID Selection:** The dropdown is automatically populated with all the unique "Star ID" entries found in your CSV file. Select the star for which you want to calculate the final RV.

- **Select Wavelength Source (”Use”):** These radio buttons allow you to choose which wavelength measurement from the CSV file to use for the Doppler shift calculation:
  - **Fit Center:** The mathematical center of the fitted model (e.g., the mean of the Gaussian). This is generally the most robust choice.
  - **Fit trough:** The wavelength corresponding to the minimum flux of the fitted model.
  - **Original trough:** The wavelength corresponding to the pixel with the minimum flux in the raw data. This can be affected by noise.
- **Compute RV:** Click this button to perform the final calculation.

The Calculation Process:

- **Data Grouping:** The code filters the CSV data for the selected ”Star ID” and then groups all measurements by the line they belong to (’Blue’, ’Center’, ’Red’).
- **Statistical Analysis:** For each of the three lines, it calculates the mean and standard error of the mean (SEM) of the chosen wavelength measurement (e.g., Fit_Center).
- **Individual RVs:** The Doppler shift formula is applied to the mean wavelength of each of the three lines to calculate three independent radial velocity measurements.
- **Final Combination:** The final RV is the mean of these three individual RVs. The final uncertainty is the standard error of the mean of the three RVs, which properly reflects the scatter between the line measurements.

Output: The final results are presented in a comprehensive summary printed to the output area. This summary includes the number of measurements for each line, the mean observed wavelengths with their uncertainties, the individual RV for each line, and the final combined radial velocity and its uncertainty. This entire summary is also saved to a .dat text file for your records.

Barycentric correction : is a process used in astronomy to account for the Earth’s motion and gravitational effects when analyzing observations, particularly those involving light or other radiation. This is a small under RV calculation in the notebook. For more information, visit (https://docs.astropy.org/en/stable/coordinates/velocities.html)

12. USING THE INCLUDED TRIAL DATA

To help you get started and validate your results, we have included a set of trial data from an M3 mask. This dataset is particularly useful because it contains four stars of interest that have been previously analyzed by the APOGEE survey, meaning their radial velocities (RVs) are well-established and publicly available. We highly encourage you to complete the full reduction workflow for this mask, calculate the RVs for these stars using the provided tools, and then compare your results to the known values from the APOGEE catalog, which are listed in the table below.

Included with the data is a log file named test_M3_7_log_2025-07-30.txt. It is important to note that this file is provided solely for the purpose of testing the ”Apply Settings from Log” functionality. The parameters within this log are not the optimal settings for reducing the M3 mask data, although they will produce reasonably good results.

We have intentionally omitted the log file containing the precise parameter settings that we used to achieve values close to those in the APOGEE catalog. This is to encourage you to explore the parameter space on your own and to avoid influencing your choices during the reduction process. By finding the optimal settings yourself and achieving accurate results, you can independently verify the effectiveness and reliability of the pipeline.

Table Column Descriptions:

- **APOGEE ID:** The official designation for each star in the APOGEE catalog, which can be used for cross-referencing.
- **center:** The radial velocity (RV) calculated by our pipeline using the center of the fitted Gaussian profile for the spectral lines.
- **center corrected:** The ”center” RV value after applying a barycentric correction to account for the Earth’s motion.
- **fit-trough:** The RV calculated using the wavelength at the lowest point (trough) of the fitted Gaussian profile.
- **fit-trough corrected:** The ”fit-trough” RV value after applying the barycentric correction.
- **original-trough:** The RV calculated using the wavelength of the pixel with the minimum flux in the original, un-fitted spectrum.
- **original-trough corrected:** The ”original-trough” RV value after applying the barycentric correction.
- **Barycentric Values:** The correction values calculated using the Astropy library’s barycentric correction tools.
- **VHELIO:** The barycentric-corrected radial velocity for each star as published in the APOGEE catalog. This is the reference value for comparison.
- **center, fit trough, and original trough (Difference Columns):** The final three columns represent the direct difference between our calculated RVs (from each respective method) and the authoritative VHELIO value from the APOGEE catalog.

									RVs Difference		
APOGEE ID	center	center corrected	fit-trough	fit-trough corrected	original-trough	original-trough corrected	Bricentric values	VHELIO	center	fit trough	original trough
2M13420954+2819531	-149.775	-149.014314	-148.593	-147.832311	-144.106	-143.3452996	0.7610662012	-149.8013286	0.7870145507	1.969017551	5.695328575
2M13420673+2820567	-142.276	-141.5243287	-142.332	-141.5803289	-132.357	-131.6053038	0.7520281694	-150.8384829	9.314154217	9.258154076	18.48148295
2M13420876+2821125	-137.576	-136.8219041	-137.309	-136.5549034	-143.221	-142.4669183	0.7544420971	-145.7561835	8.934279335	9.201280007	2.535183454
2M13421266+2820420	-143	-142.2372528	-142.256	-141.4932509	-136.22	-135.4572355	0.7631112023	-145.7591971	3.52194433	4.265946224	9.539197129

Table 1: This table contains the RV data of the APOGEE catalog and our measured RVs using the pipeline