EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

FORS Spectroscopy EDPS-GUI tutorial

1 rev. 20260811

Issue 1

Date Aug 11 2026

Prepared:	Danuta Dobrzycka	Aug 11 2026	.....
	Name	Date	Signature
Approved:			.....
	Name	Date	Signature
Released:			.....
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1	11/08/2026	all	First public release for version 5.8.7

This page was intentionally left blank

Contents

1 Introduction

1.1 Scope

This document describes how to reduce FORS SPECTROSCOPY data with the `edps-gui` (Graphic User Interface), the dashboard of the ESO Data Processing System (`EDPS`), which is the recommended interface to reduce data from ESO telescopes. Details on the FORS SPECTROSCOPY data reduction stream and how to configure the reduction to meet specific scientific needs are also given.

For a more extensive documentation on the `edps-gui` itself, consult the dedicated manual here.

For a description of the FORS SPECTROSCOPY pipeline itself, consult the pipeline manual available at: https://www.eso.org/sci/software/pipe_aem_table.html.

Note: this tutorial refers to:

- FORS SPECTROSCOPY instrument pipeline named `fors`, version 5.8.7.
- FORS SPECTROSCOPY workflow: `fors.fors_spec_wkf`
- `EDPS` version 1.5.7.
- `edps-gui` version .9.1.

1.2 What is `EDPS`?

The ESO Data Processing System (`EDPS`) is a framework to run ESO's data processing pipelines and it is meant to eventually replace the previous ESOReflex environment. The general principles of `EDPS` have been described by Freudling, Zampieri, Coccato et al. [2024, A&A, 681, A93]. Please refer to that paper if you have used `EDPS` for research resulting in a scientific publication.

Each of ESO's data processing pipeline consists of a series of standalone programs called *recipes*. Each recipe is designed to process certain type(s) of input data. The processing of these input data typically requires a range of auxiliary files such as calibration files. `EDPS` is designed to select appropriate input data for the different recipes of a pipeline, and execute them in sequence. This is done by specifying for each pipeline the workflow for organizing data and executing the recipes. Then, the workflow can be used to process a set of data fully automatically.

1.3 Main concepts

`EDPS` is an environment designed to execute the recipes of an instrument pipeline according to a series of instructions. The main concepts in `EDPS` are:

- **Workflow** and **reduction cascades**. A workflow is a series of instructions designed to reduce data with an instrument pipeline in potentially multiple ways, by carrying on a sequence of tasks. Each workflow can define multiple reduction cascades, depending on the scientific needs. For example, the same workflow

can be used to process data following different strategies that trigger different reduction steps (e.g. in one strategy flux calibration can be omitted) or different end-points (e.g., combine different science exposures, or stop after the reduction of individual exposures without combining them). Each of these "strategies" defines a "reduction cascade".

- **Task**, **jobs**, and **recipes**. A task is an element in the workflow that performs a given step of the data reduction cascade. Tasks are often associated to a recipe of the underlying instrument pipeline. A jobs is a work unit in a processing environment, that runs a recipe on a set of input data with a set of recipe parameters. A single task can generate several jobs: for example, a "bias" task, can generate multiple jobs, each of the running the bias recipe on a different set of input files.

- **Dataset**. A dataset is a collection of files, that are needed to perform the data reduction as specified by the workflow. It consists, for example, of one or more science files plus the calibrations needed to process them. In `EDPS`, datasets have a hierarchical structure, which highlights the connections between the various files and tasks (e.g., task A is an input to task B).

- **Target** and **Target category**. The "target", or the "target task" is the end point of the reduction cascade. When specifying a target, `EDPS` will process all and only the files needed to execute it. For example, if my target is "science", and the science files need the bias files, `EDPS` will process only the biases that have been selected to process those science files; then it processes the science using the product of the bias reduction. However, if my target is bias, then `EDPS` will process all and only the bias files, regardless they are not used by any science. In this case, `EDPS` does not process the science, as it has already reached the end reduction point (e.g., process all biases). The "Target category" is a group of targets that have similar purposes. For example, the target category "science", includes all the tasks that deliver final scientific products, the target category "qc1calib" includes all and only the tasks that processes calibrations (e.g., bias, flat fields, standard stars).

1.4 Installation

1.4.1 Prerequisites

Prerequisites for a well functioning installation of `EDPS` and `EDPS`-gui are:

- Recent Firefox or Chrome browser, Python 3.11 or higher (but there are issues with Python 3.14).

- At least one ESO pipeline with EDPS workflow should be in your system. To install the desired ESO pipelines, follow the instructions in the ESO pipelines pages. NOTE: the `apptainer` installation method is currently not supported. After the installation, the esorex command must be in the path. To test whether the installation was successful, type

  ```
  esorex --recipes
  ```

  A list of available recipes should appear.

- Install `graphviz`, `fv`, and `ds9`, which have to be included in the system path (defining aliases not enough). On linux, `Graphviz` can be easily installed via:

```
sudo apt install graphviz (Debian, Ubuntu)
sudo dnf install graphviz (Fedora)
```

Check the `Graphviz` webpage for installation instructions for other OS.

`fv` and `ds9`, are optional. To install them, follow the instructions in corresponding webpages. You can test whether these three packages are installed and their paths are correctly set by typing on a terminal:

```
dot -V
fv  -version
ds9 -version
```

1.4.2 Installation steps

To install EDPS follow these steps:

- Create a new Python virtual environment and activate it:

  ```
  python3 -m venv edpsgui
  . edpsgui/bin/activate
  ```

  Make sure the python3 version is 3.11 or higher, but not 3.14.

- Install the required packages:

  ```
  pip install --extra-index-url \
      https://ftp.eso.org/pub/dfs/pipelines/repositories/stable/src \
      edps edpsgui edpsplot adari_core
  ```

To run the `edps-gui` type from a terminal (with the active environment):

```
edps-gui
```

Important note. The first time `edps-gui` is executed, you will be asked to specify the directory where the reduction products (fits files and quality plots) will be stored. The default location is `$HOME/EDPS_data`. During the first execution, a configuration file named `application.properties` will also be saved in the directory (newly created) `$HOME/.edps`.

2 Reducing demo data

Follow this procedure to quickly reduce FORS SPECTROSCOPY demo data. We assume that the `EDPS`, `edps-gui`, the FORS SPECTROSCOPY pipeline and its associated demo data are installed in your system. For general instructions on how to install `EDPS` and the pipeline, see Section 1.4 or please visit: https://www.eso.org/sci/soft

2.1 Setting the workflow

Proceed as follows:

1. If not done already, activate the `EDPS` virtual environment, defined during installation (Sect. 1.4).
2. Start the `edps-gui` dashboard by typing:

   ```
   edps-gui
   ```

   The `edps-gui` dashboard will start in a browser window (Figure 1).

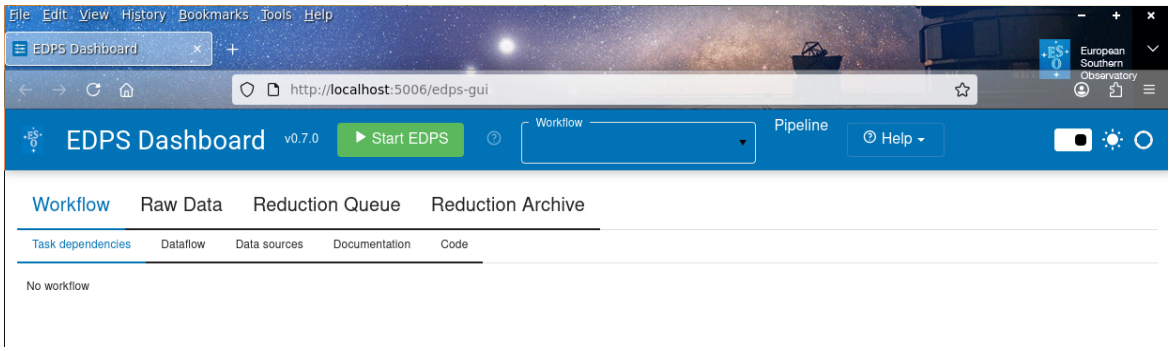

Figure 1: The empty `edps-gui` Dashboard; the underlying `EDPS` engine has not yet been started and no workflow has been loaded.

3. Optionally, before starting `EDPS`, one can specify new settings by pressing Help –> Settings (Figure 2).

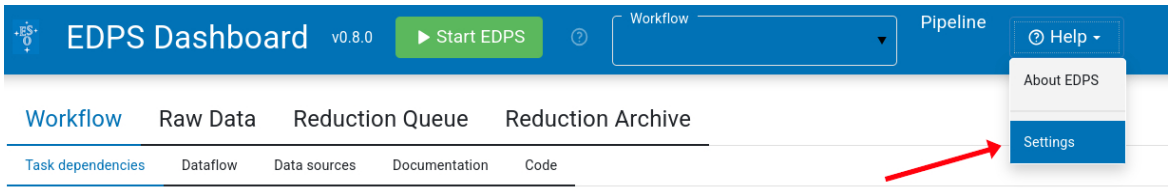

Figure 2: The “Help” –> “Settings” menu.

4. On the browser window with the dashboard, press the button ‘Start EDPS‘.
5. Choose the `fors.fors_spec_wkf` workflow from the list in the ’Workflow’ field. The workflows offered in this selector depend on the installed pipelines. The graphic workflow representation will appear as in Figure 4.

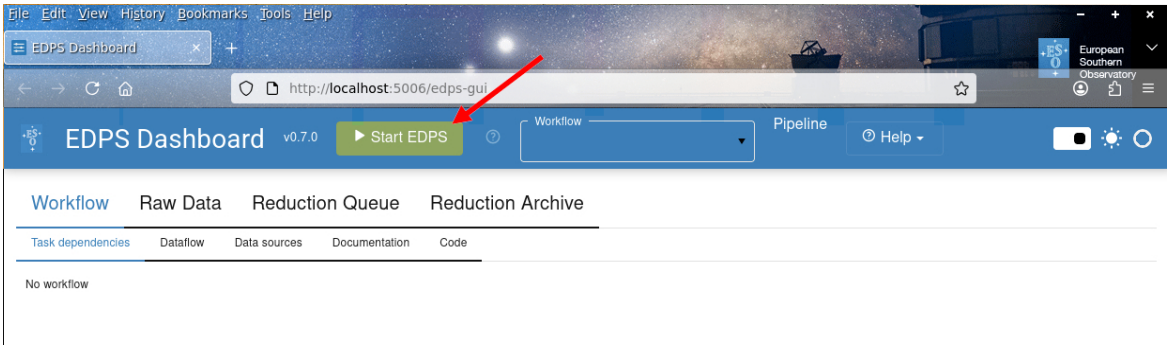

Figure 3: The “Start EDPS” button.

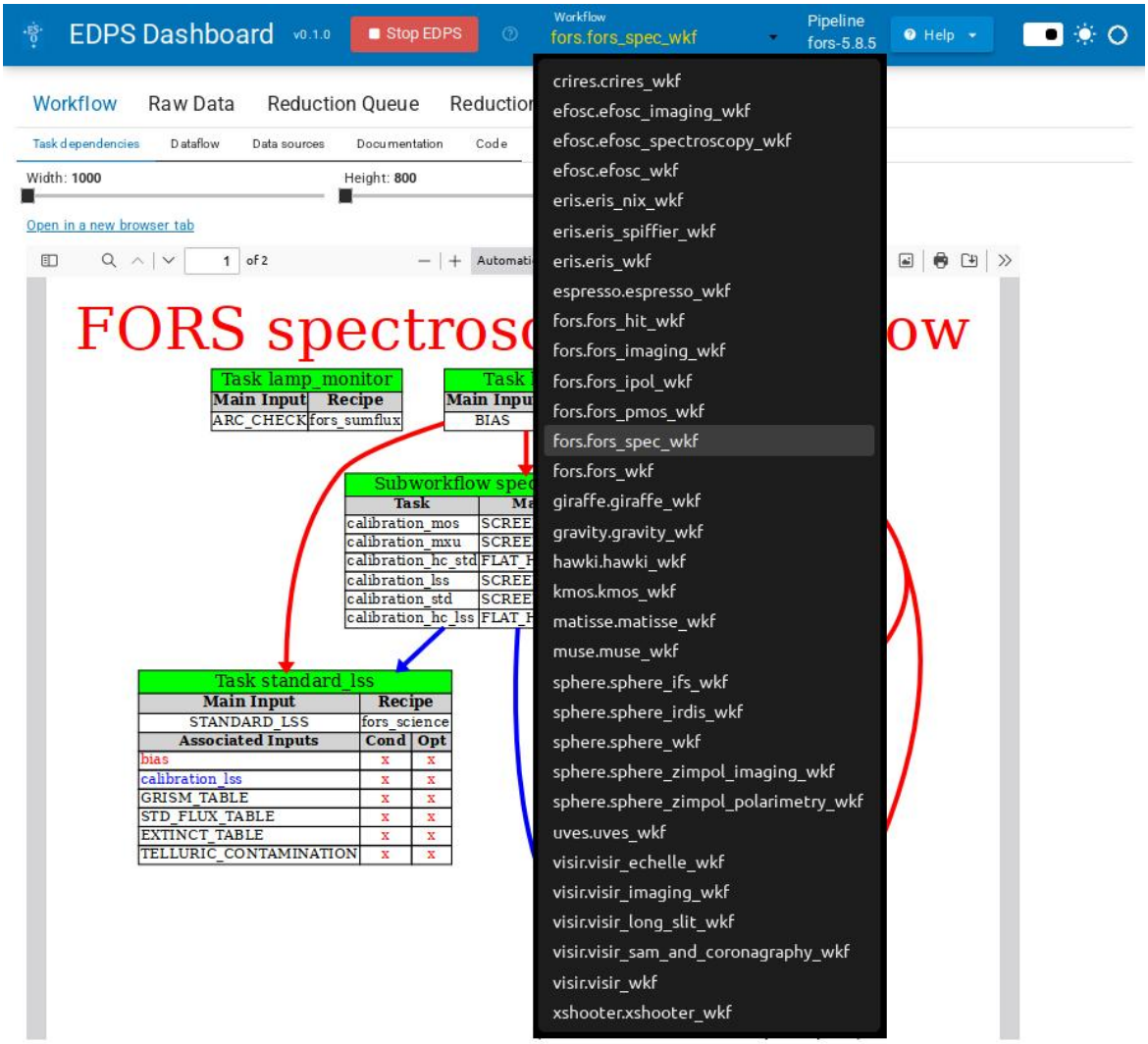

Figure 4: The `edps-gui` with the FORS SPECTROSCOPY workflow loaded.

2.2 Selecting the input data

1. Press ‘Raw Data‘ to enter the corresponding tab, as in Figure 5.

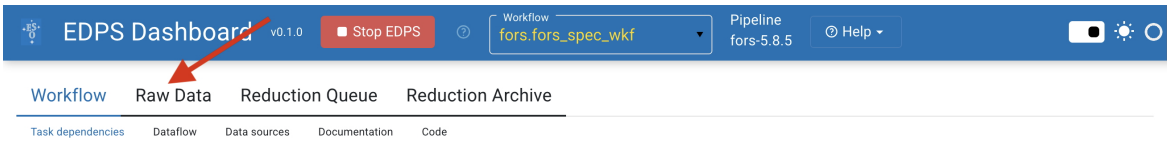

Figure 5: How to select RAW data Tab.

2. Press ‘Select Inputs‘. A selection window will appear that allows to select data that are stored on a local disk (Figure 6).

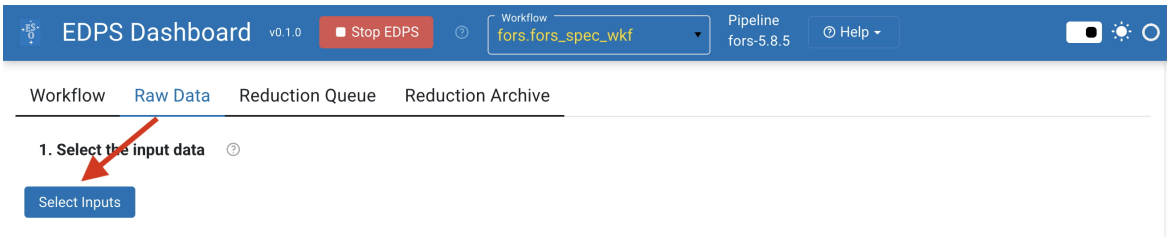

Figure 6: How to select input data.

3. (Optional). Select the reduction target, configure the workflow parameter and specify the association preferences. These steps are optional. For more information see Section **??**.

4. Press ‘Create Datasets‘. A list of datasets appears, one line for each set of science data (Figure 7).

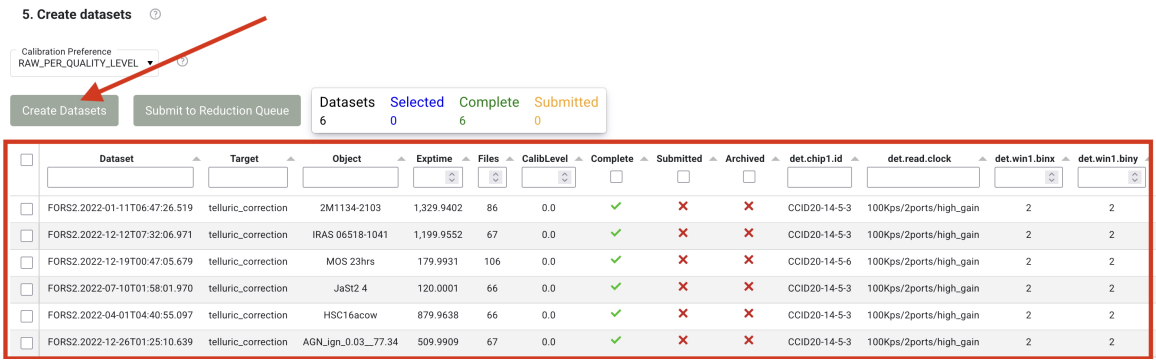

Figure 7: How to inspect the input data directory to create datasets.

5. Choose the datasets that should be processed (Figure 8)

   and send them to the data reduction queue by pressing ‘Submit to Reduction Queue‘. Note that this action does not start the reduction automatically.

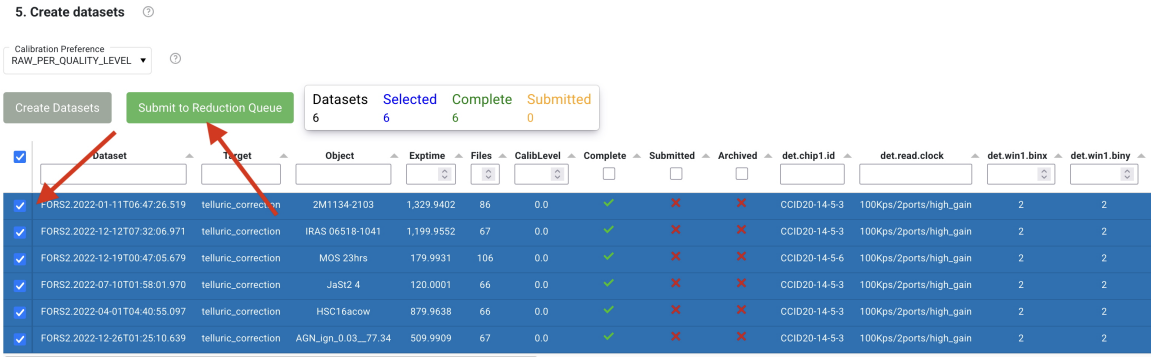

Figure 8: How to send the selected datasets to the Reduction Queue for processing.

2.3 Start the reduction

1. Press the ‘Reduction Queue‘ tab (Figure 9).

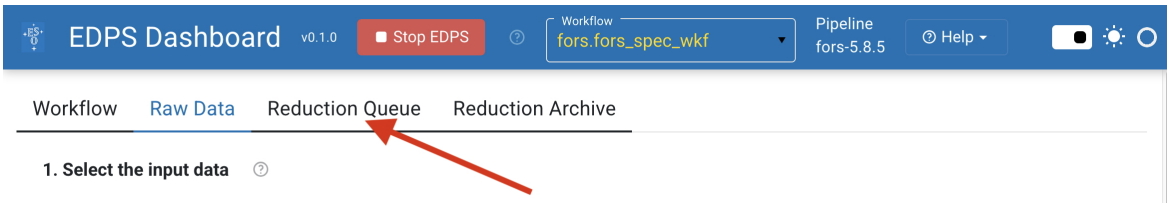

Figure 9: How to select Reduction Queue tab.

2. Select the datasets you’d like to reduce.

3. (Optional). Configure the workflow and recipe parameters by pressing the wheel button  to open the configuration editor. See Section 4.2 for more information on the configuration editor.

4. Press the ‘Reduce‘ button (Figure 10). The selected data will now be processed with the configured parameters.

Congratulations! You reduced your first data with the `EDPS` dashboard! All the reduced data are saved in the EDPS_data directory specified when executing `edps-gui` for the first time.

2.3.1 Quality plots

It is possible to inspect the information on each job, such as quality plots showing the products (for the most important jobs and products), the list of inputs and output files, the recipe parameters and logs. All the information on the job processing can be inspected from the ‘Reduction Queue‘ window. While quality plots are produced only for completed jobs, the other relevant information is available also for failed jobs.

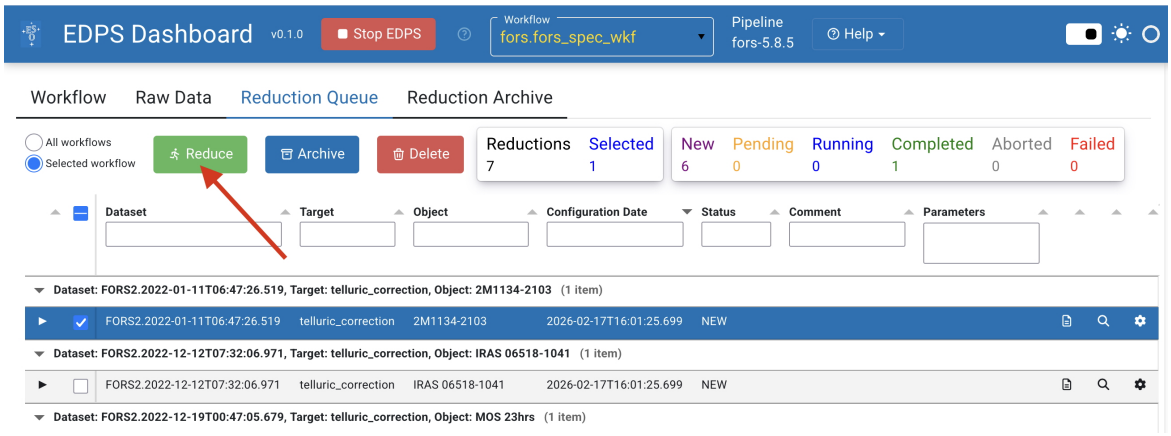

Figure 10: Reduce.

The information associated for the main product can be inspected by pressing the magnifying glass symbol at the right side of each dataset. To inspect the information associated to other individual jobs (e.g., calibrations), proceed as follows:

- Expand the desired dataset by pressing the black arrow on its left. The list of jobs will appear with the associated status (COMPLETED, RUNNING, PENDING, MISSING, ABORTED, FAILED)
- Press the magnifying glass symbol at the right side of the job you want to inspect.

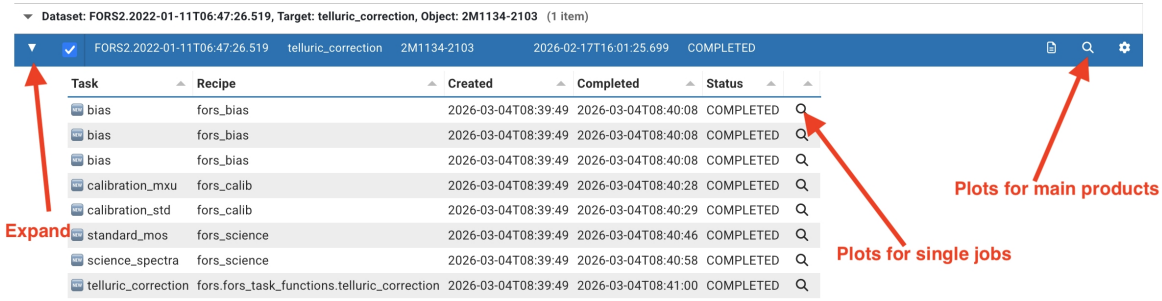

Figure 11: How to look for job information from the Reduction Queue tab.

2.4 Exporting the final products

Completed reductions can be 'Archived' (i.e. declared 'completed' because no more work is needed) and removed from the Reduction Queue. Additionally, even if all products for all tasks are saved in the EDPS_data directory, the most important products can be 'exported' to a desired location.

To do so, proceed as follows:

1. In the 'Reduction Queue' tab, select the dataset and the dataset for which you want to export the final products, and press the 'Archive' button.

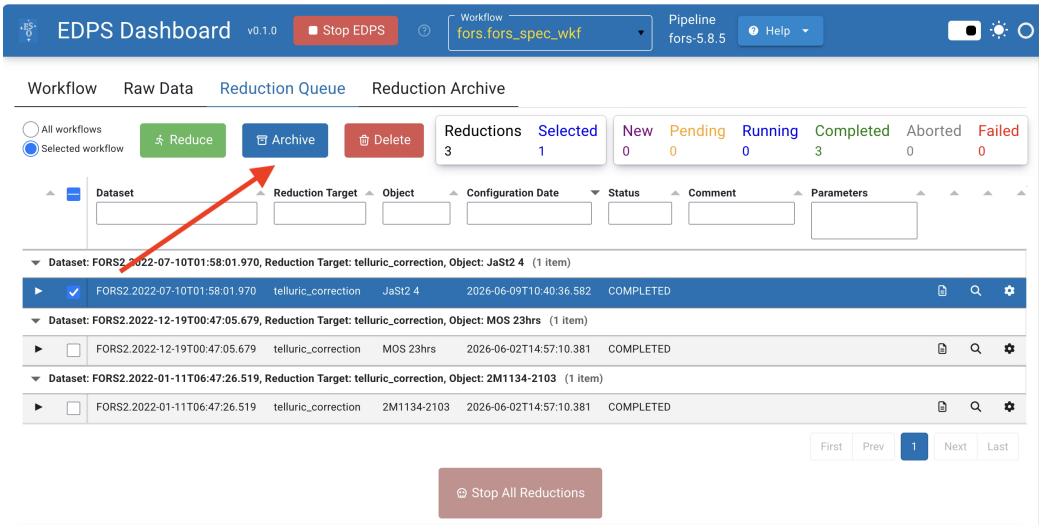

Figure 12: How to archive a completed reduction from the Reduction Queue tab.

2. Go in the Reduction Archive tab and click on the 'Export' button. A new tab window appears where you can indicate the directory you want to copy your final products; finally press "Export" to copy the data.

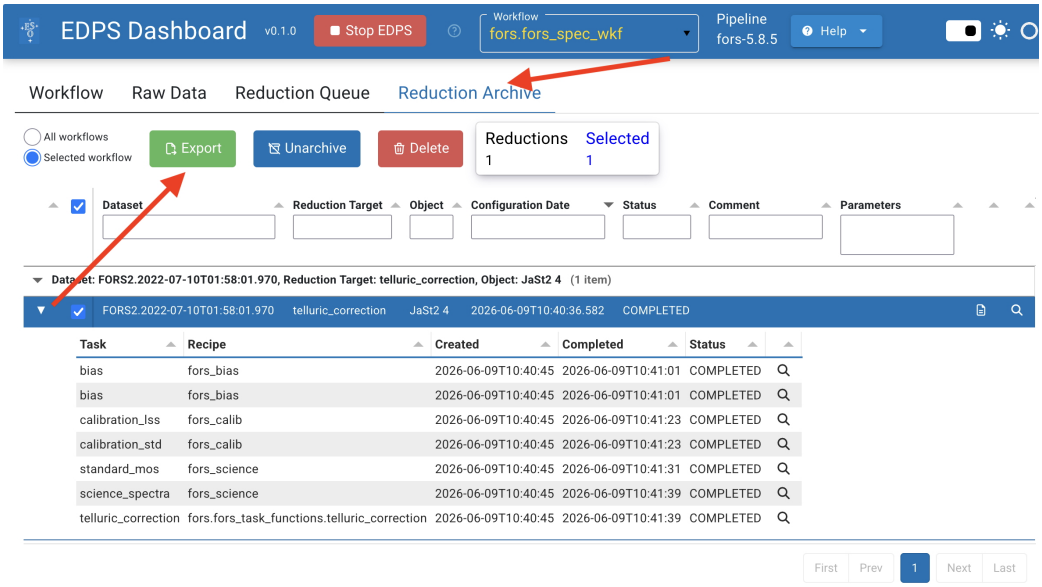

Figure 13: The reduction archive tab. This table contains all the different configurations of datasets that are declared "finished" and removed from the Reduction Queue. From this page, the user can export the most important files into a desired local directory.

Exported products are organized by 'DATASET' (named as the first scientific exposure of the dataset), and 'TIMESTAMP' (time of start of reduction)

The final products saved in the specified directory are:

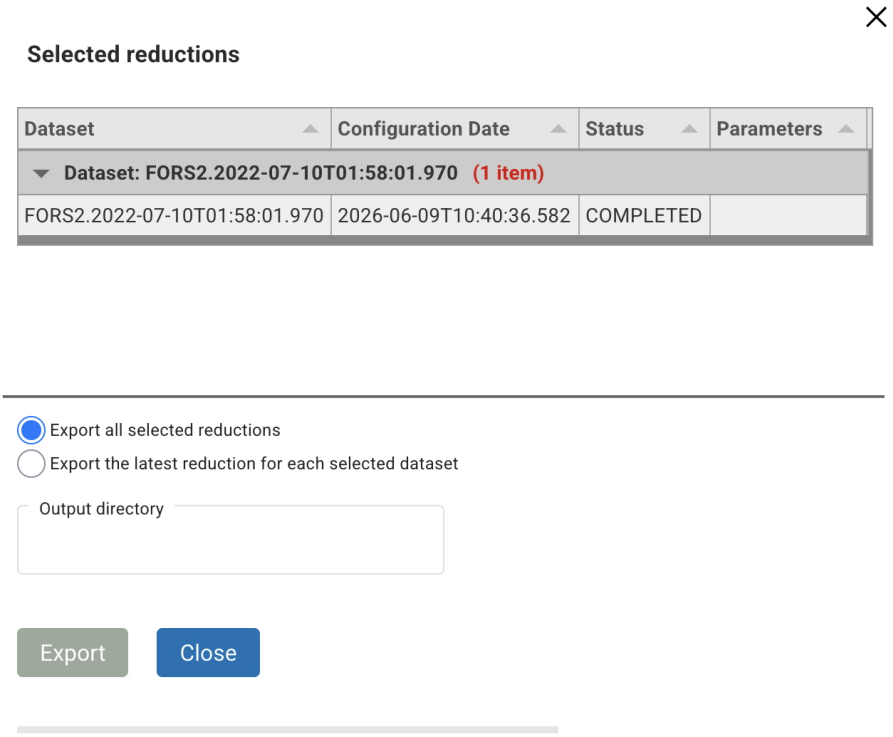

Figure 14: The EXPORT dialogue window, where the user can decide which reduced configuration to save and where.

- SPECTRUM_LSS/MOS/MXU_2D_... - 2-dimensional science frame, sky-subtracted and flux-calibrated.
- SPECTRUM_LSS_... - 1-dimensional extracted, flux-calibrated spectra (Phase3 compliant).

3 The FORS SPECTROSCOPY data reduction flow.

FORS is the visual and near-UV FOcal Reducer and low dispersion Spectrograph for the Very Large Telescope (VLT) of the European Southern Observatory (ESO). Two versions of FORS have been built and mounted at the VLT, but in April 2009, FORS1 was removed to make room for X-shooter, so only FORS2 is now in operation. FORS is designed as an all-dioptric instrument for the wavelength range from 330 nm to 1100 nm and provides an image scale in the standard mode of 0.25 arc sec per pixel with the readout mode (2×2 binning).

FORS2 offers three spectroscopic modes: long-slit spectroscopy (LSS) using a mask with $6.8'$ long slits of different widths, and multi-object spectroscopy with movable slit blades (MOS, slit length about $20''$) and with masks (MXU, arbitrary slit length, curved and tilted slits possible). The spectrophotometric standard stars for flux calibration are always taken in MOS mode, where the slitlets are used to create a long slit of $5''$ width at the position of the long-slit used for the science data or at the center of the field-of-view for multi-object spectroscopy.

FORS2 has several volume-phase holographic grisms, for which the response depends on the position on the CCD. This requires specific steps during the flux calibration, which are handled correctly by the pipeline.

The Longitudinal Atmospheric Dispersion Corrector corrects the effects of atmospheric refraction up to an airmass of about 1.5, thereby enabling observations with arbitrary angles on sky without flux losses due to atmospheric refraction.

The FORS2 supports the correction of telluric absorption using the molecfit algorithm.

This document describes reduction of the data taken in any of the spectroscopic modes of FORS1 and FORS2 (for simplicity, the instrument is often addressed as FORS). It does not cover the other modes, which are described in different tutorial documents.

The overall data flow of the FORS SPECTROSCOPY pipeline is displayed in Figure 15.

The reduction cascade is organized in tasks, which represent well-defined steps in the process. Tasks can be grouped inside sub-workflows. Each task runs a recipe; the detailed description of the algorithms, input, outputs and recipe parameters used in each recipe are available in the pipeline manual. Here, we present only the description of most important features.

The `fors.fors_spec_wkf` `EDPS` workflow is designed to execute the tasks that deliver the final reduced data cube for each dataset. It can be either the product of a single exposure, or the combination of multiple exposures. Only calibrations needed by the selected the scientific exposures are processed.

It is possible to set `EDPS` to perform the data reduction until a certain step of the reduction chain (e.g. to reduce only standar stars, or only flat fields). This is done by specifying the desired tasks in the field **Select reduction target** of the **Raw Data** tab.

The reduction steps of the `fors.fors_spec_wkf` workflow are listed below. Before starting the reduction, the parameters of the recipes associated to each task can be configured by pressing the button  close to each dataset configuration. See for more info on the configuration editor 4.2

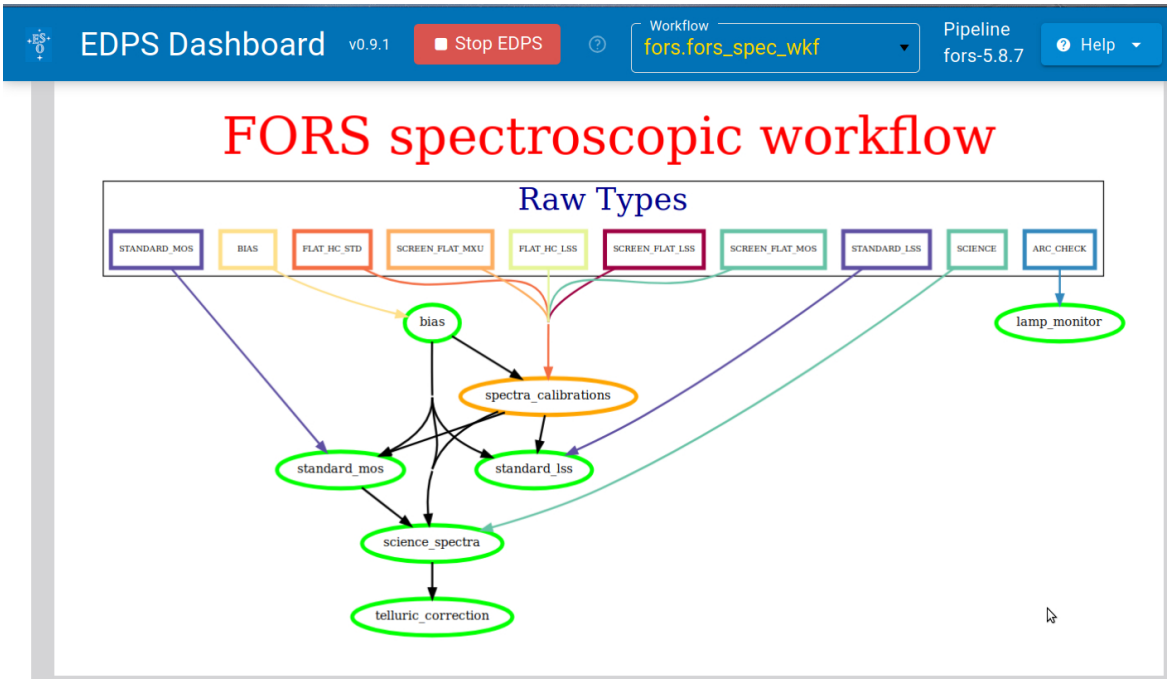

Figure 15: The data reduction cascade of the FORS SPECTROSCOPY workflow.

3.1 Create Master Bias

Task: **bias** Recipe: **fors_bias**

In this step the combined MASTER_BIAS is produced out of the input raw frames. The bias raw frames are acquired with an exposure time of 0 seconds and a closed shutter. They thus record only the signal that is added during the read-out of the CCD to avoid negative numbers. A sequence of 5 or 20 bias frames is taken the day following the observations as part of the FORS2 calibration plan (the calibration plan was changed in November 2014; Since then, 20 instead of 5 bias frames are taken per setting). Using the full bias frame instead of just the pre-overscan allows the user to correct for potential offsets between the bias level on the detector and the pre-overscan regions.

The same MASTER_BIAS product per setup is further used in reduction of the LSS, MOS and MXU data.

Figure 16 demonstrates how the pipeline products can be examined on the example of the product of the task: **bias**, recipe **fors_bias**. Further, the Figure 17 displays an example of the quality plot for the MASTER_BIAS. The image should be uniform, without visible flux gradient. You can also display another extension, IMAGE.ERR containing estimated error per pixel. See Section 4.3 for more information on how to improve the product if needed.

3.2 Create Master Flat Field, Dispersion Solution and Correction for Spatial Distortion

Tasks: **calibration_lss**, **calibration_mos**, **calibration_mxu**, **calibration_std** Recipe: **fors_calib**

In one step, the input screen flat raw calibration frames: SCREEN_FLAT_LSS/MOS/MXU together with the

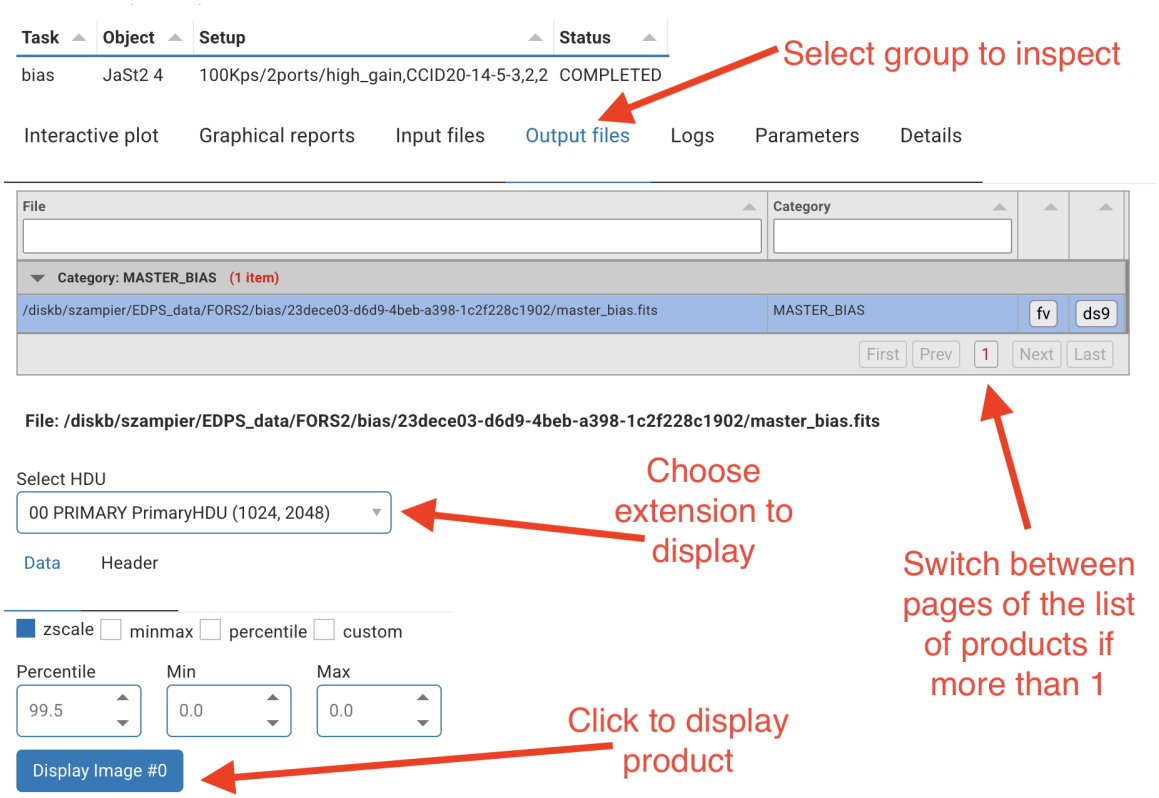

Figure 16: Examining product of the task: **bias**, recipe: **fors_bias**.

arc lamp raw frame: LAMP_LSS/MOS/MXU are used to create a normalized master flat-field, a 2D dispersion solution, correction for spatial distortion, and to determine the slit limits.

The exposure of the arc lamp(s) He, HgCd, and Ar (at the lowest spectral resolution with grism 150I) and in addition the Ne lamp (at higher resolution) are acquired together with the screen flat field data to ensure that they are all recorded at the same mask position.

The task **calibration_std** produces similar master calibrations used to process spectrophotometric standard star data - the 5 arcsec slit width MOS observations.

The tasks **calibration_hc_lss** and **calibration_hc_std** also run the **fors_calib** recipe but only reduce the FORS spectroscopic calibration data used for monitoring.

Main Products:

Not all output frames listed here are always produced. Some are never created in case of LSS or LSS-like data. The LSS-like data are obtained in the MOS instrument mode when all the slits are aligned. This kind of data are processed as a single long slit spectrum. The product categories will contain the acronym LONG before the instrument mode tag: for instance, DELTA_IMAGE_MOS will become DELTA_IMAGE_LONG_MOS.

- MASTER_SCREEN_FLAT_LSS/MOS/MXU - master screen flat field (for verification only); comparing it to the normalized master flat allows to see if all slitlets for MOS/MXU have been found

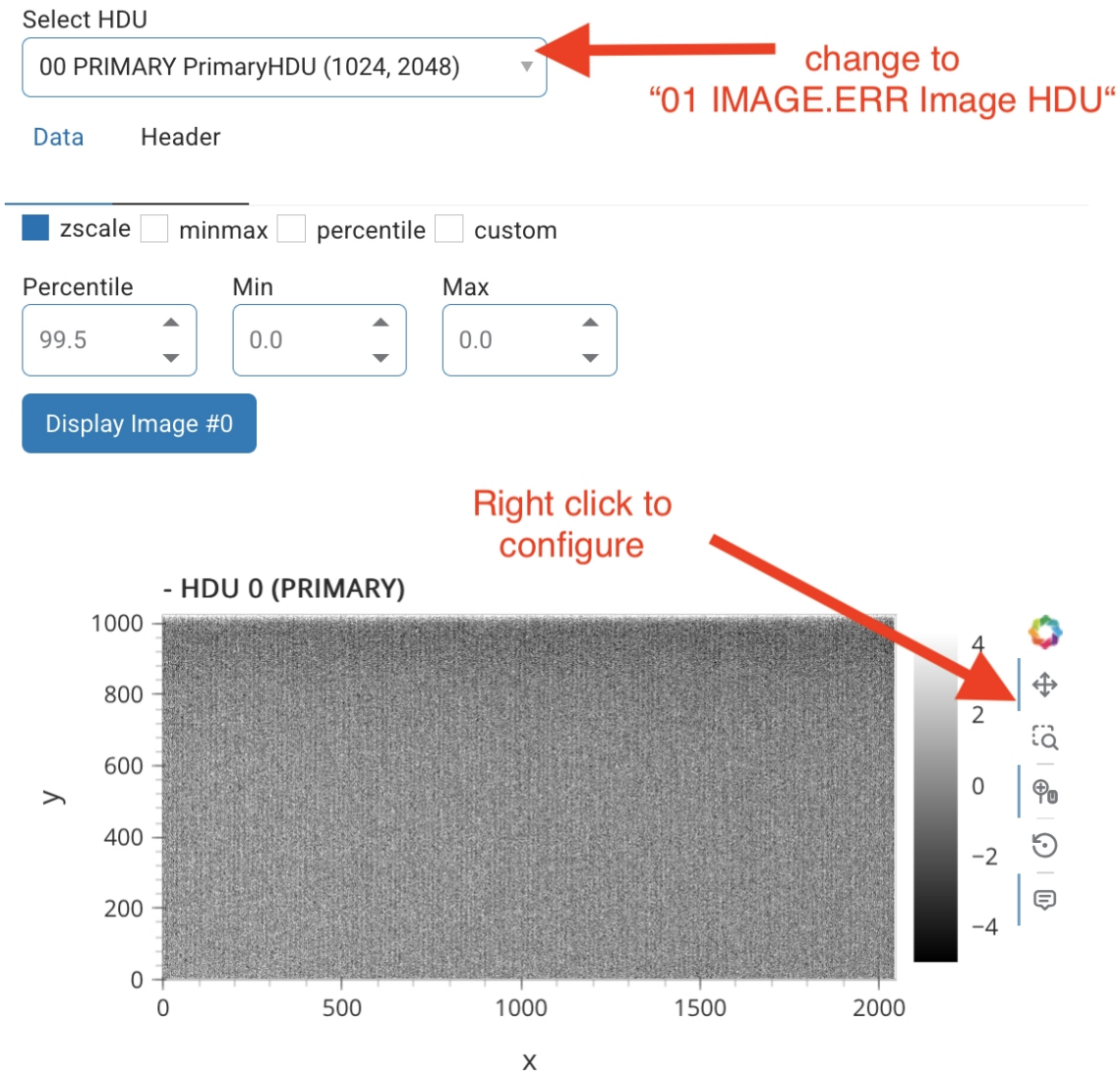

Figure 17: The quality plot of the MASTER_BIAS.

- MASTER_NORM_FLAT_LSS/MOS/MXU - normalized master flat field, derived by dividing the master screen flat by its smoothed version
- DISP_COEFF_LSS/MOS/MXU - table containing the wavelength calibration polynomial coefficients
- DISP_RESIDUALS_LSS/MOS/MXU - residuals of each wavelength calibration fit (in pixels)
- DISP_RESIDUALS_TABLE_LSS/MOS/MXU - table containing different kinds of residuals of a sample of wavelength calibration fits
- FLAT_SED_LSS/MOS/MXU - image containing the spectral energy distribution of the flat; each image

row corresponds to the SED (Spectral Energy Distribution) of one slit ordered as in the SLIT_LOCATION_MXU table; the SED profile is wavelength calibrated using the polynomial of the central row of the illuminated region

- SLIT_LOCATION_LSS/MOS/MXU - table containing the slit positions on the CCD and on the reduced lamp image
- WAVELENGTH_MAP_LSS/MOS/MXU - wavelength map image with the same size as the CCD, where each pixel is assigned the value of the wavelength at its center
- SPATIAL_MAP_MOS/MXU - spatial map image (MOS/MXU only), which has the same size as the CCD, where to each pixel is assigned the value of its distance (in CCD pixels) from the top edge of the spectrum it belongs to
- SPECTRAL_RESOLUTION_LSS/MOS/MXU - spectral resolution table containing the mean spectral resolution for each reference arc lamp line
- CURV_COEFF_MOS/MXU - table (MOS/MXU only) containing coefficients of the polynomials used to fit the spectral curvature
- CURV_TRACES_MOS/MXU - table (MOS/MXU only) containing the y CCD positions of the detected spectral edges
- REDUCED_LAMP_LSS/MOS/MXU - reduced lamp image (for verification only), i.e. rectified and wavelength calibrated arc lamp image. This is the result of applying the extraction mask derived from the flat field and arc lamp exposures to the input arc lamp exposure itself. This image is useful to get an immediate feeling of the goodness of the computed extraction mask.
- DELTA_IMAGE_LSS/MOS/MXU - deviation from the linear term of the wavelength calibration polynomials

In this step, few properties of the master calibrations should be checked : quality of the wavelength calibration, distortion correction (not applicable to LSS and LSS-like data), the flat field combination and normalization. It is recommended to create and inspect "Graphical reports" (for "Both" "Raw data" and "Reduced data") and the "Output files" links among the quality plots.

Figure 18 shows an example of the wavelength-calibrated arc lamp frame (REDUCED_LAMP_MXU product), while Figure 19 the example of the "Graphical report" FORS_reduced_lamp_mos. In these plots the arc lamp lines should run straight from top to bottom without any empty rows between them. Particular attention should be given to lines at the blue and red ends of each spectrum, where the polynomial fit is more sensitive to small variations of the signal. Some arc lines may show gaps due to the placement of the slitlets, but empty rows without any lines point towards problems with the detection of the arc lamp lines. For LSS and LSS-like data the arc lines extend from top to bottom without any slitlet marks.

Residuals between predicted and detected arc line position (the DISP_RESIDUALS_LSS/MOS/MXU product) should generally be below 0.5 pixel. If they show systematic variations the polynomial degree used to fit the dispersion relation may be too low (or in rare cases too high). If the scatter appears very large one should zoom in, because there are often only a few outliers and the majority of the residuals are within ±0.5 pixels.

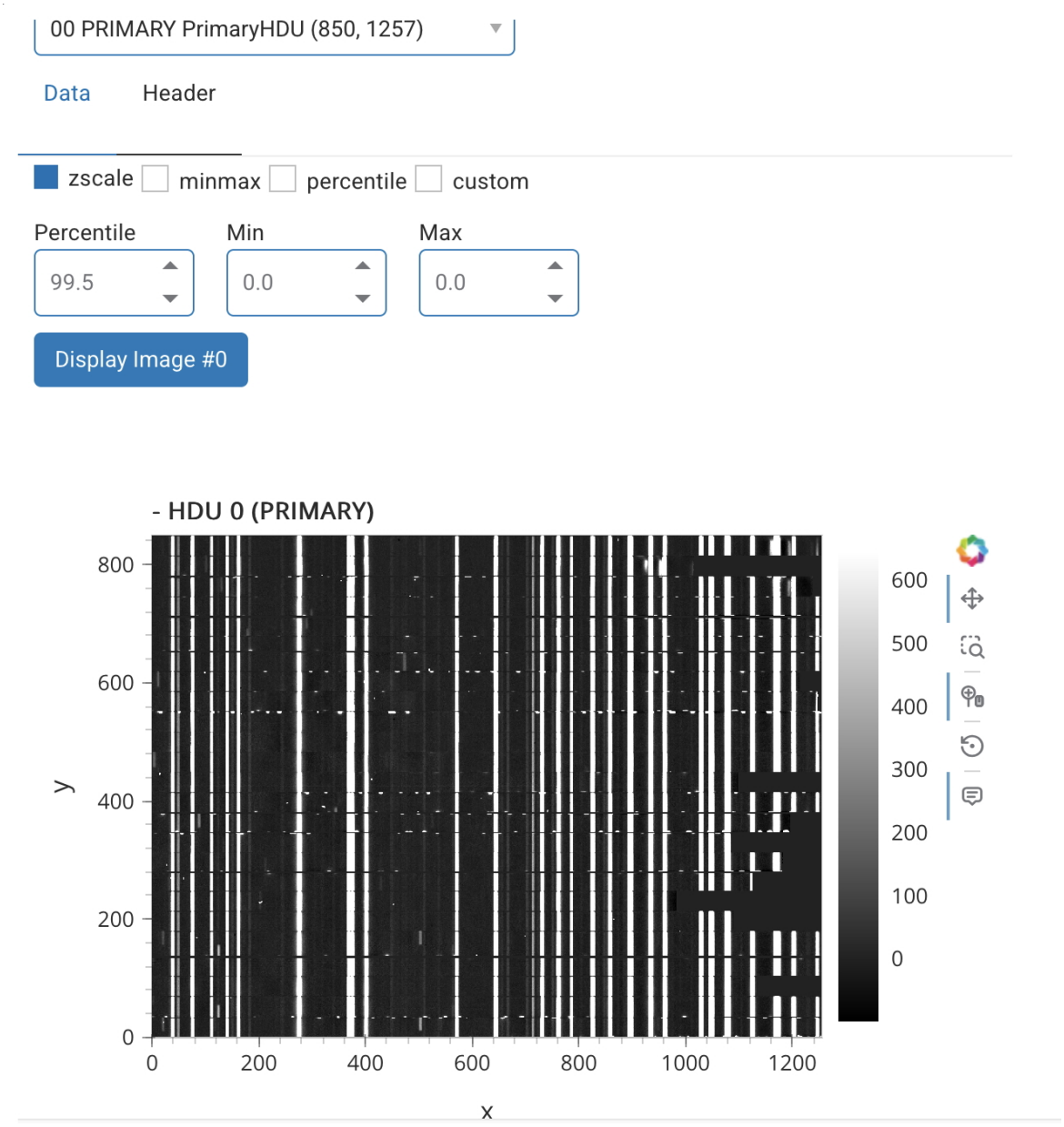

Figure 18: Example of the quality plot of the REDUCED_LAMP_MXU product of the task: **calibration_mxu**, recipe: **fors_calib**.

The spatial map (SPATIAL_MAP_MOS/MXU product) has as pixel values the distance of a pixel from the bottom of the respective slitlet. The regions of the slitlets should not be strongly curved nor should regions of different slitlets overlap with each other.

Figure 20 shows an example of the MASTER_NORM_FLAT_MXU product. The master flat should have the same number of slitlets as the first raw flat and their areas should also be identical. All slitlets should be detected and there should be no spurious detections (e.g. one slitlet detected as several). For LSS and LSS-

Figure 19: Example of the "Graphical Report" for the task: **calibration_mos**, recipe; **fors_calib** reduced_lamp_mos.png.

like data the illuminated region along the y-axis has to be identical between the REDUCED_LAMP_LSS and the MASTER_NORM_FLAT_LSS. Otherwise part of the exposed area is lost. In the frame, depending on the normalization method residual gradients may be validly present (e.g. for `sradius:` -1 as recommended for LSS data).

See Section 4.4 for more information on how to improve the products if needed.

3.3 Compute Response Function

Tasks: **standard_lss**, **standard_mos** Recipes: **fors_science**

The observed spectrophotometric standard stars are stars with a well known flux distribution. They are usually observed during twilight within ±3 days of the science data. The data are processed to create the instrument response curves. They may be later used to flux-calibrate science data in case there is no master response curve determined for the particular grism (affects mostly observations before January 1, 2025. After that date, the master response curves exist for most of the grisms). The recipe **fors_science** integrates the observed flux, corrected for gain, exposure time and atmospheric extinction, over the intervals indicated in the STD_FLUX_TABLE corresponding to the observed star (the `HIERARCH ESO OBS TARG NAME` keyword is matched between the observation and the entries in the table). The reference flux values for the star are then divided by the integrated

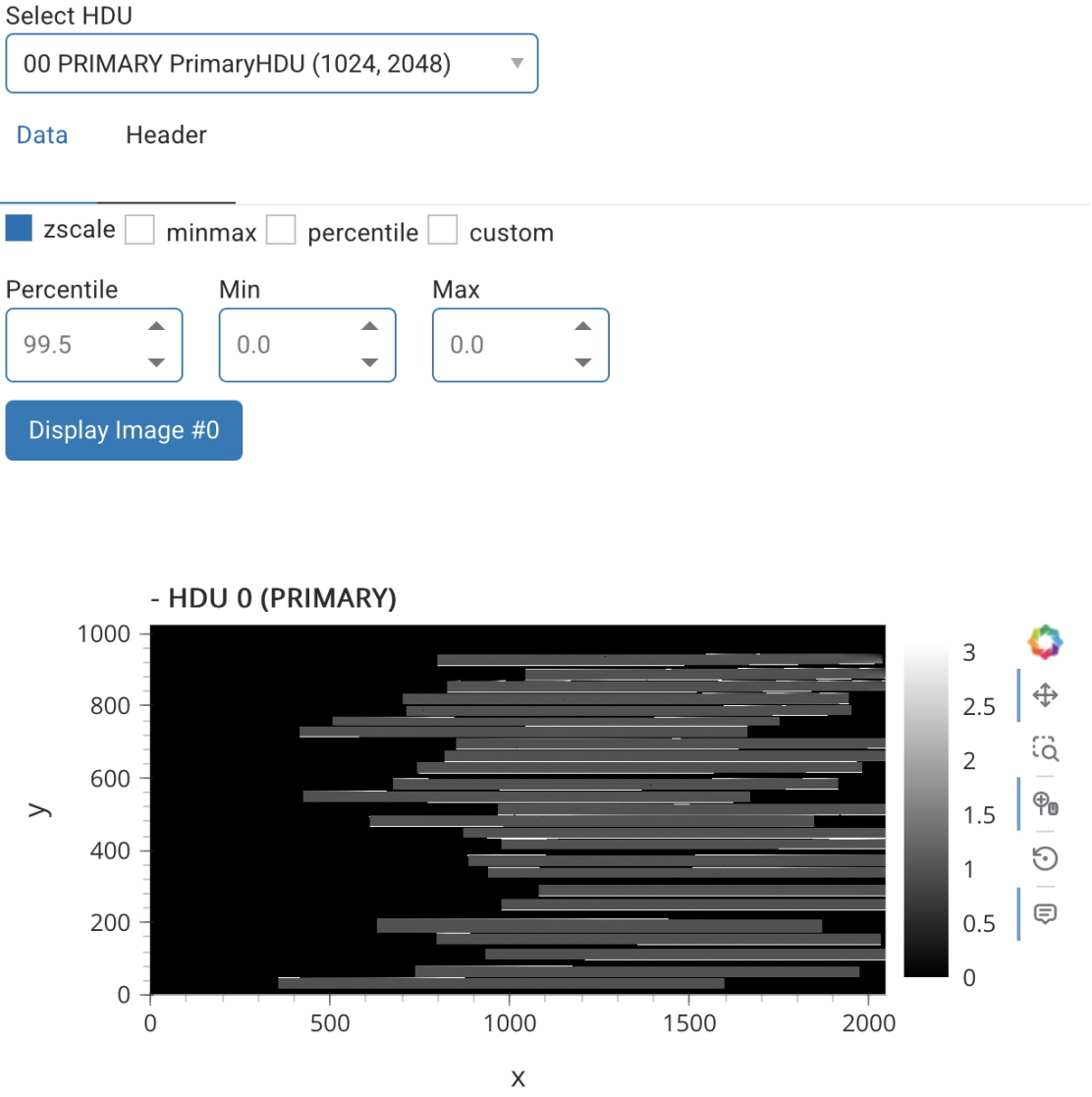

Figure 20: Example of the quality plot of the MASTER_NORM_FLAT_MXU product of the task: **calibration_mxu**, recipe: **fors_calib**.

observed flux values to create the so-called raw response. Features like strong stellar lines or regions of telluric absorption are masked before a polynomial or a spline fit is performed to define the final response curve.

Product:

- SPECPHOT_TABLE - table with efficiency and response curves. It can be used as an input to the **fors_science** recipe when the flux-calibrated products are created.

In this step, the quality of the response curve fit should be checked.

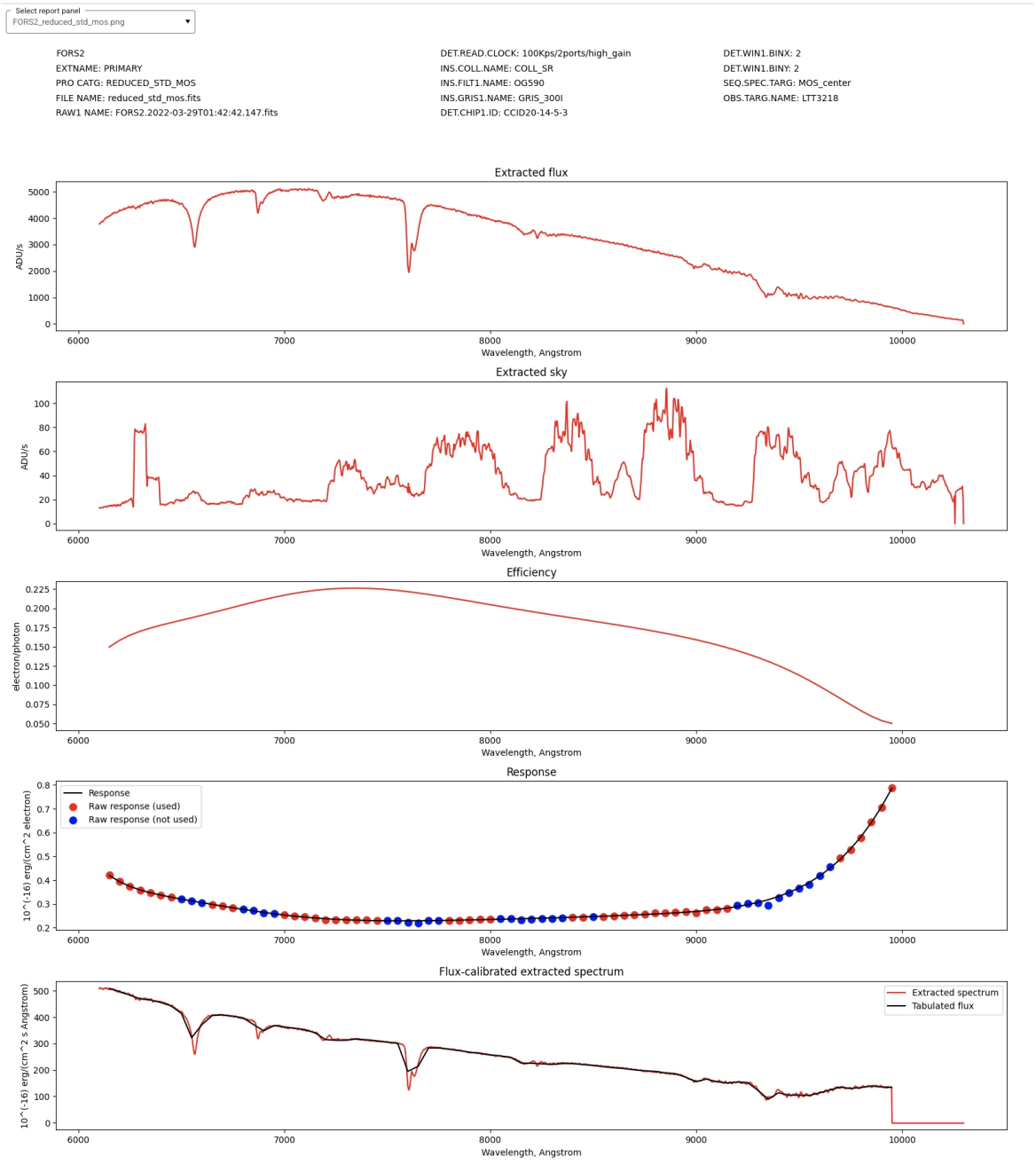

Figure 21: Example of the "Graphical report" reduced_std_mos.png of the task: **standard_mos**, recipe: **fors_science**.

Figure 21 is an example of the created "Graphical report" for the relevant task: **standard_mos**, recipe: **fors_science**. The extracted standard star spectrum should show no jumps or sky emission lines. Strong gradients due to order separating filters are valid but may cause problems with the fit of the response curve.

In the "Response" panel, the dots show the raw response (ratio of reference spectrum and observed spectrum

integrated over same bins as reference spectrum) and the blue line shows the corresponding fit. Blue dots are masked and not used for the fit.

In the bottom panel the red line marks the observed standard star spectrum calibrated with its own response curve and the black curve indicates the reference data (blue points were masked during the fitting of the response). Differences between the black and the red lines indicate a problem with the flux calibration, usually features on a scale smaller than the bins of the reference data. Differences between the red and black curves may indicate problems with inter- and/or extrapolation, but may also be valid, for instance due to different amounts of telluric absorption or to differences in resolution between the reference spectra and the observed data.

Raw and fitted response - the blue curve (fit) should follow the red curve (unmasked raw response) closely. If the fit deviates from the data in the masked regions in unexpected ways (blue dots) the user may try to remove `stellar_absorption` and/or `telluric` from the parameter `resp_ignore_mode` and mask only smaller regions via `resp_ignore_points`. Alternatively or in addition one may change the fit method and/or degree via `resp_fit_degree` (polynomial degree) or `resp_fit_nknots` (number of spline knots). One should keep in mind that interpolating across a larger region (e.g. covered by telluric lines) provides at best a probable response curve, but may also show strong deviations. This is even more true for extrapolations, e.g. if a response curve has telluric regions at its edges.

Flux-calibrated standard star spectrum should coincide with the tabulated standard star flux (blue dots were ignored during the response fit). Deviations at the very blue end may results from problems with atmospheric extinction and deviations at wavelengths above 6000 Å may point to problems with telluric absorption.

Keep in mind that occasionally there may be no STD_FLUX_TABLE associated to the observations of a flux standard star.

3.4 Reduce Science Data

Task: **science_spectra** Recipe: **fors_science**

In this task the recipe **fors_science** is run to apply sky subtraction and extract the spectra. The science observations get flux calibrated using the master response curve (preferred if available) or the instrument response curve derived from the standard star observation. If neither standard star observation nor master response curve is available, then the science observation will not be flux-calibrated.

Products (the filenames for the other modes are similar but with the MOS suffix replaced by MXU, LONG_MOS or LSS as appropriate):

1-dimensional extracted spectra (Phase3 compliant)

- REDUCED_IDP_SCI_MOS - flux-calibrated spectra
- SCI_MOS_TELLURIC_CORR - flux-calibrated spectra, corrected for telluric absorption. The first extension is identical to the corresponding REDUCED_IDP_SCI_MOS product, the second extension contains information on the telluric correction parameters, and the third extension provides the telluric corrected spectrum (column `cflux`). Please note that the wavelength in the third extension (column `lambda`) is in μm, not in Å.

1-dimensional extracted spectra (image format, REDUCED_*, created only if spectra are identified and can be extracted). The individual spectra are provided as rows in a FITS file. The correspondence between these rows and the 2-dimensional frames and/or slit identifications can be obtained from OBJECT_TABLE_SCI_MOS.fits. All extracted spectra have the same format.

- REDUCED_SCI_MOS - spectra
- REDUCED_ERROR_SCI_MOS - error of spectra
- REDUCED_FLUX_SCI_MOS - flux-calibrated spectra
- REDUCED_FLUX_ERROR_SCI_MOS - error of flux-calibrated spectra
- REDUCED_SKY_SCI_MOS - sky spectra

2-dimensional wavelength calibrated and distortion corrected frames (MAPPED_*). For LSS data the distortion is corrected only if an appropriate GLOBAL_DISTORTION_TABLE was provided (this table currently does not exist for grism 150I (all filters), grism 200I (no filter), grism 300I (no filter), grism 300V (filters GG375, GG435), grism 600R (filter GG435), and grism 600I (filter FILT_465_250).)

- MAPPED_ALL_SCI_MOS - 2-dimensional SCIENCE frame without sky subtraction
- MAPPED_SCI_MOS - 2-dimensional SCIENCE frame, sky-subtracted
- MAPPED_FLUX_ALL_SCI_MOS - 2-dimensional SCIENCE frame, flux calibrated, without sky subtraction
- MAPPED_FLUX_SCI_MOS - 2-dimensional SCIENCE frame, sky-subtracted and flux-calibrated
- MAPPED_SKY_SCI_MOS - 2-dimensional frame with fitted sky background

2-dimensional frames, neither wavelength calibrated nor distortion corrected (UNMAPPED_*)

- UNMAPPED_SCI_MOS - (not for LSS) 2-dimensional SCIENCE frame, sky- subtracted, neither wavelength calibrated nor distortion corrected
- UNMAPPED_SKY_SCI_MOS - 2-dimensional frame with fitted sky background, neither wavelength calibrated nor distortion corrected

table with position information for detected spectra

- OBJECT_TABLE_SCI_MOS

If sky alignment is requested (skyalign ≥ 0) the following products are provided in addition to the ones listed above:

- DISP_COEFF_SCI_MOS - dispersion coefficients after adjusting to sky line positions

- SKY_SHIFTS_SLIT_SCI_MOS - shifts in wavelength derived from sky line positions (for MOS-like data). For long-slit and LSS-like MOS (LONG_MOS) data this is SKY_SHIFTS_LONG_SCI_MOS.
- WAVELENGTH_MAP_SCI_MOS - 2-dimensional frame with pixel value=wavelength of pixel.

All products <type>_FLUX_* are created only if appropriate flux standard star observations for the upper chip are provided and the standard star flux table is available (<type> being REDUCED or MAPPED).

In this step it is good to check the quality of the sky subtraction and spectrum extraction. Note that some products contain spectra in ADU/sec, i.e. not flux-calibrated.

When displaying the MAPPED_SKY_SCI_MOS/MXU product - the wavelength-calibrated, rectified frame after sky subtraction, the detected spectra ranges (slits) should be horizontal. In particular, for the LSS data your spectra should be horizontal. This is most easily done by zooming onto a spectrum, selecting the full range in wavelength but only a small range along the y-axis. In rare cases the use of the GLOBAL_DISTORION_TABLE results in tilted instead of horizontal spectra in the MAPPED... data. The cause for this behaviour is still unclear.

The extracted science spectra should not show strong residuals of sky lines. A quick check on sky subtraction can be made by examining the sky subtracted frame MAPPED_SCI_MOS/MXU/LSS. The spectra should have a generally smooth look, and only appear to be noisier in those regions where bright sky lines were subtracted. The best way to ensure that the sky was subtracted optimally, at least at the positions of the objects to extract, is to check that the residual noise is compatible with the statistical error associated to the extracted object spectra. The extracted spectra are contained in the REDUCED_SCI_MOS/MXU/LSS image (one extracted spectrum for each row). Their error spectra (at a 1 - σ level) are contained in the REDUCED_ERROR_SCI_MOS/MXU/LSS image. The regions of the extracted spectra corresponding to a (bright) sky line may include a few noisier points, which deviation from the spectral continuum should (almost) never pass the 3 - σ deviation. If this condition is fulfilled, the sky subtraction is probably as good as it can get. The sky spectra extracted from the modeled sky frames in exactly the same way as the object spectra from the sky-subtracted frames can be found in REDUCED_SKY_SCI_MOS/MXU/LSS. Note that if the barycentric correction is applied to the products, the wavelengths are re-computed to match the desired reference system. No interpolation is done on the spectrum itself, only wavelengths are changed. This means that spectra of the same target might be defined at different wavelengths, depending on the velocity correction. This has to be taken into account when combining spectra for further analysis.

3.5 Performing Telluric Correction

Task: **telluric_correction** Recipes: **fors_molecfit_model**, **fors_molecfit_calctrans**, **fors_molecfit_correct**

The telluric correction is performed only in case of spectra extending beyond 6800 A. Currently, this has been implemented only for the LSS mode data but is going to be extended for other modes in the near future. To enable this correction the workflow parameter `Apply_telluric_correction` must be set to anything but `false` and this is done automatically by the workflow for the data with `WAVELMAX` larger than 6800 A. In case the Reduction Target **telluric_correction** was selected by the user but the correction was not applied, the **science_spectra** products are copied over to the **telluric_correction** data products directory.

The way the correction works is that for each observation, the 1D spectrum with the highest signal-to-noise ratio is selected and used as reference spectrum to be fitted by the recipe **fors_molecfit_model**. This recipe computes

the abundances of various molecules taking into account also the instrument set-up. Then, the best fit parameters obtained in **fors_molecfit_model** are used to construct the atmospheric transmission for each of the extracted spectra in the observation. This step is performed by the recipe **fors_molecfit_calctrans**. Last, each spectrum is corrected for telluric absorption using the appropriate transmission by the recipe **fors_molecfit_correct**.

Note: although the science spectra share the same reference spectrum used to determine the best fit parameters, each individual science spectrum must have its own correction because they might differ in wavelength extension (e.g., in the case of MOS or MXU observations).

In general, for the FORS data, it is sufficient to fit only O2 and H2O molecules.

Note: Only 1-dimensional spectra in Phase3 format are corrected, not the various 2D frames produced by the FORS2 science recipe.

3.6 Static Calibration Data

In addition to calibration frames taken regularly, the FORS pipeline also uses static calibration tables related to spectroscopy.

1. MASTER_LINECAT - This table contains the reference wavelengths (in Å) for the arc lamps and grism used.

2. GRISM_TABLE - This table contains grism-specific parameters, like the dispersion in Å/pixel, the start and end wavelength, the polynomial degree to be used for the wavelength calibration, fit parameters for the response curve, etc. There are separate GRISM_TABLEs per instrumental setup.

3. GLOBAL_DISTORTION_TABLE - This table provides the correction of the spatial distortion for long-slit spectroscopy. It exists for most, but not all grisms, and is an optional input.

4. STD_FLUX_TABLE - This spectrophotometric standard star table file is needed to determine the instrumental response. It contains the reference flux of the stars together with regions of strong stellar absorption that are by default ignored when fitting the response curve.

5. TELLURIC_CONTAMINATION - This table contains, for each grism affected by telluric absorption, the wavelength regions that are by default ignored when fitting the response curve.

6. EXTINCT_TABLE - This table describes the atmospheric extinction at the Paranal observatory. It is needed for the creation and application of the response curve.

7. EOP_PARAM - The Earth Observation Parameter table contains the Earth Orientation Parameter as a function of time (MJD-OBS). This file is needed for the barycentric correction.

8. STD_FLUX_CATALOG - The flux standard star catalog files provide the reference data (model spectra) for the 7 flux standard stars, that are used to determine the response.

9. FIT_POINT_CAT - The fit_point catalogs provide the wavelength points per grism, filter and star. These points are used to fit a response curve to the raw response (ratio of reference to observed spectrum).

10. WAVE_INCLUDE - Table with wavelength regions for telluric absorption. It contains, for each combination of grism and filter, the wavelength regions to be used by the **fors_molecfit_model** recipe to fit the telluric absorption.

11. MOLECULES - This table contains the list of the molecules used by the **fors_molecfit...** recipes. A description of these tables is given in the FORS Pipeline User Manual.

4 Overview of all the data reduction configuration options

4.1 Selection of most appropriate calibrations

By default, EDPS associates raw calibrations to the reduction process. It is also possible to use pre-processed calibrations (a.k.a. master calibrations) if available, in order to speed up the reduction. The preference can be specified in the Raw Data tab, before creating the datasets.

Possible values of the Calibration Preferences are:

- **raw_per_quality_level**: At equal quality of reduction, association of raw calibrations is preferred. This is the default.
- **master_per_quality_level**: At equal quality of reduction, association of master calibrations is preferred.
- **raw**: Association of raw calibration is preferred, despite the quality of results.
- **master**: Association of master calibration is preferred, despite the quality of results.

When master calibrations are used, the reduction step needed to process raw calibrations are not executed. The reduction then moves directly to the process of scientific exposures.

For example, if reduction speed for a quick check is preferred over a high quality reduction, one can select "master". In this case, old master calibrations are associated even if there are raw calibrations closer in time (and therefore more likely to ensure better quality products).

The quality level that the selected calibrations deliver is indicated close to each dataset in the ‘Raw input‘ tab, under the colum ‘CalibLevel‘. CalibLevel=0 indicates that calibrations that follow the rules of the instrument calibration plans have been selected. The higher the number, the poorer the quality of the products.

4.2 Configuration of parameters: the configuration editor

The data reduction of each dataset can be configured according to the scientific needs using an appropriate configuration editor.

The EDPS workflows contain two types of parameters and they both have default values that can be modified to improve the data reduction.

- **Workflow parameters** (for some workflows only) are global and they are applied to the entire workflow. They are accessible both in the ‘Raw Data‘ tab, prior to the creation of a dataset, and in the ‘Reduction Configuration‘ editor, in the ‘Reduction queue‘ tab. Note: some workflow parameters were already configured before creating the dataset and sending it to the reduction queue. Here, they can be changed again. Please, note that the parameters have an effect only on the files that are already in the dataset. If one specifies a parameter that should include extra files in the dataset (e.g., the inclusion of more calibrations), files are not added and the reduction might fail. If you need to change a parameter that modifies the dataset content, please go back to the Raw data tab and create a new dataset.

- **Recipe parameters** are specific to the individual recipes and can be configured per task. They are accessible in the ‘Reduction Configuration‘ editor, in the ‘Reduction queue‘ tab.

This editor allows to configure the data reduction for a given dataset by specifying workflow and recipe parameters.

Note: some workflow parameters were already configured before creating the dataset and sending it to the reduction queue. Here, they can be changed again. Please, note that the parameters have an effect only on the files that are already in the dataset. If one specifies a parameter that should include extra files in the dataset (e.g., the inclusion of more calibrations), files are not added and the reduction might fail. If you need to change a parameter that modifies the dataset content, please go back to the Raw data tab and create a new dataset.

To open the editor, click on the wheel button  next to the dataset you desire to configure the reduction for. A window with the configuration editor appears as shown Figure 22.

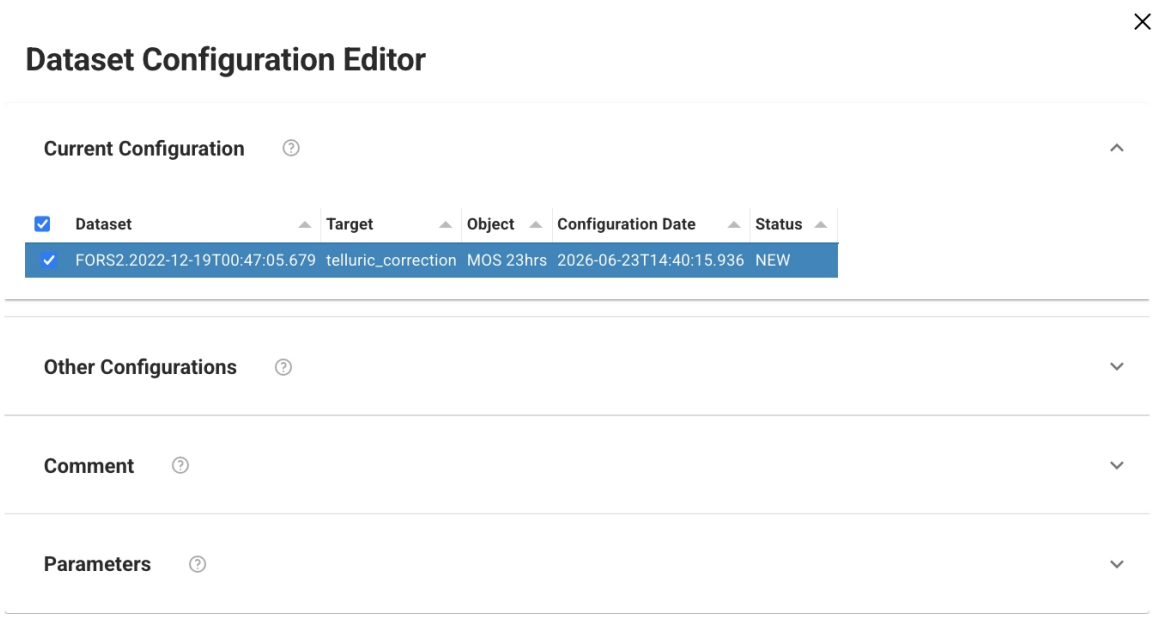

Figure 22: The Reduction Configuration editor. It contains 4 sections, that indicate the current configuration, list of other configurations to set, comments to insert, and the parameters to modify.

The editor is divided into 4 parts, which can be accessed pressing the corresponding expansion arrow.

- **Current configuration.** It indicates the name of the selected configuration for a given dataset (Figure 23).
- **Other configurations.** It allows to specify other configurations, to which the changes shall be copied to (Figure 24).
- **Comment.** It allows to specify a comment to describe the configuration. It is possible to append or replace a comment (Figure 25). Comments can be changed on all configurations. It is possible to save the comment for the current configuration only, or for all the selected configurations.

- **Parameters.** A window as in Figure 26 appears.

  The window allows to:

  – Select the parameter set. A pre-determined list of workflow parameters and recipe parameters for a given use case. For the majority of the cases, the "science" parameter set can be used.

  – Edit the workflow parameters. These are parameters that regulates the reduction strategy, e.g. whether to use a given calibration or not, or to trigger a certain reduction step. Note that if the changes imply that some files not in the dataset are needed, the reduction might fail. In case, go back to the raw data tab, edit the workflow parameters there, and recreate the datasets.

  – Edit the recipe parameters. These are parameters associated to the recipe of a given task. Note: the same recipe parameters can be configured differently for the tasks that run the same recipe. Default parameters are shown (albeit some parameters can be dynamic, e.g. 'EDPS' changes their value depending on the type of input data).

  Change the values according to the needs and then select whether to save it to the current or the selected configurations. Note, complete configurations cannot be modified, new configurations will be automatically created instead.

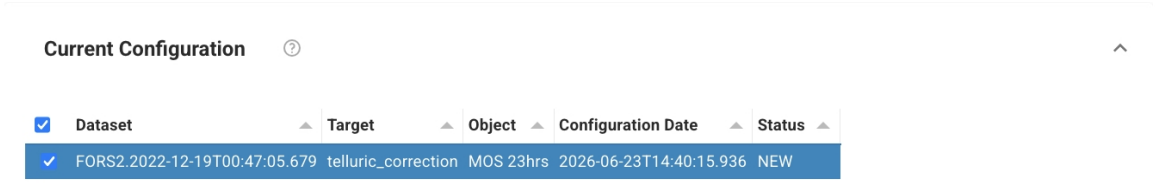

Figure 23: The first part of the Reduction Configuration Editor, that indicates the selected configuration.

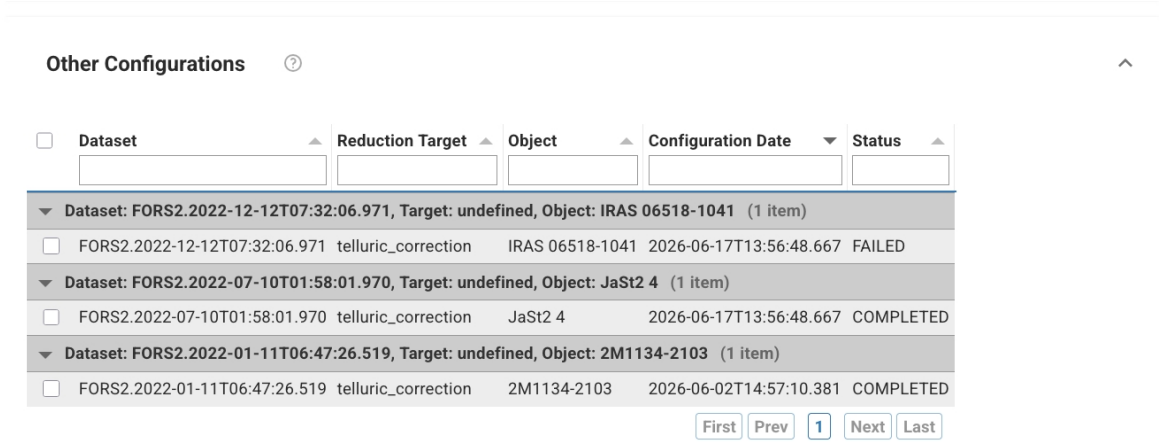

Figure 24: The second part of the Reduction Configuration Editor, that indicates other configurations for which we'd like to apply the changes.

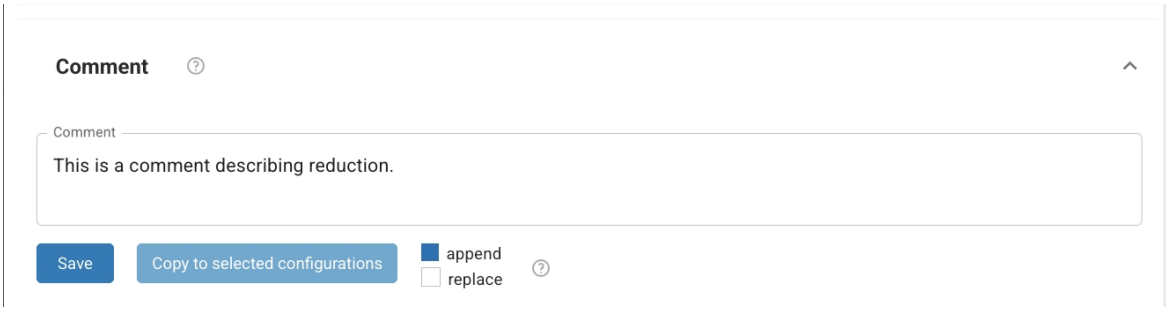

Figure 25: The third part of the Reduction Configuration Editor, that allows to specify a comment to the selected configurations.

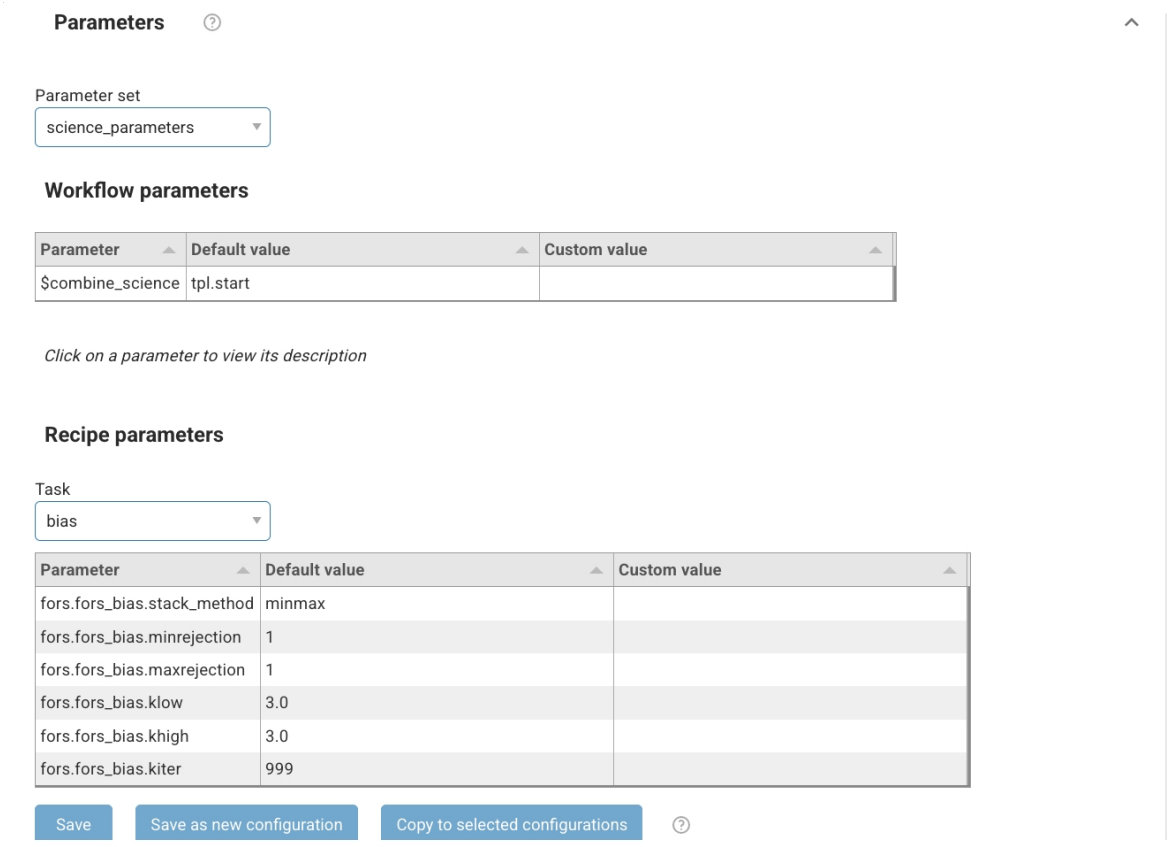

Figure 26: The fourth part of the Reduction Configuration Editor, that allows to specify the parameters sets and the recipe parameter per task. These settings can be applied to the "Selected Configuration" (Fig. 23) or to the "Other Configurations (Fig. 24).

4.2.1 Apply the same parameters to multiple datasets and configurations

A common user case is to apply the same set of optimized parameters to other datasets or other configurations. This is possible with the **Other configurations** tab of the configuration editor. To do so, proceed as follows:

- From the Reduction Queue tab, select the dataset/configuration that has the parameters you want to adopt for other reductions.
- Open the configuration editor by clicking on the weel-button on the right-hand side, at the end of the line of the selected configuration.
- If necessary, edit the parameters.
- On the **Other configurations** tabs, select the datasets or configurations you want to reduce with the same parameters of the "Current configuration". Only datasets/configurations present in the Reduction Queue can be selected.
- Go on the Parameters tab of the configuration editor and click on the button **Copy to selected configurations**. Selected configurations that were labelled with "NEW" will be updated, selected configurations that were labelled with "COMPLETED/FAILED/ABORTED" will not be updated, but "NEW" configurations will be created instead.
- If you have edited paramters on the 3rd step, please note that the original "Current configuration" will not be updated by pressing the button **Copy to selected configurations**; only those selected in the "Other configurations" tab will be updated. If you want the edited parameters to be copied also to the "Current configuration", then press the "SAVE" button (available only if the Current configuration is labelled as NEW) or "Create new configuration" button (in this case the Current configuration is not updated, but a "NEW" configuration is created instead).

The newly created or updated configurations are now ready for reduction.

4.3 Producing optimal Master Bias

Figure 17 shows an example of the quality plot for the MASTER_BIAS Output file created with the recipe parameters as in Figure 26. The image should be smooth, without visible flux gradient. If it is not, check the input raw files. Exclude the visible "bad" ones if there are any.

4.4 Configuring processing of calibrations and improving results

The tasks: **calibration_lss**, **calibration_mos**, **calibration_mxu** and **calibration_std** run the same recipe **fors_calib**, so they share the same recipe parameters.

The parameters can be divided into three groups - dealing with the normalization of the combined flat, with the creation of the master flat, and with the wavelength calibration, distortion.

1. Flat field normalization

   Following recipe parameters deal with the normalization of the combined flat field along the dispersion and/or spatial axis. The fitting or smoothing is performed on 1-dimensional data created by collapsing the 2-dimensional frame along the other axis.

- `sradius` and `dradius`: provide the half-width of the smoothing boxes along the spatial and dispersion axis, respectively. The smoothing is performed by calculating the median across the full box width. A value greater than -1 for `dradius_aver` defines the half-width of a smoothing box along the dispersion axis, using the average instead of the median. This is useful for spectra covering wavelength below 4000 Å, where the detector shows significant pixel to-pixel variations that should be kept in the normalized flat field to enable their correction in the science data.
- `s_degree`: provides the degree of a polynomial to be fit to the average variation along the spatial axis.
- `d_knots`: provides the number of knots for a spline to be fit to the average variation along the dispersion axis.

2. Creation of a master flat

- `stack_method`: offers various methods to combine the flat fields: a simple sum (sum), an average (mean), a median (median), or an average with κ - σ clipping (ksigma). The default value sum is usually fine.
- `kiter`: number of iterations (only for `stack_method` = ksigma)
- `ksigma`: value for κ (only for `stack_method` = ksigma)
- `slit_ident`: If activated the pipeline tries to match the detected slitlets with slit position information from the header (not applicable to long-slit data). Activating this matching may improve the accuracy of the preliminary spectra detection. If the slit identification fails the parameter is forced to false.

3. Wavelength calibration, distortions

- `startwavelength` and `endwavelength`: define the maximum wavelength range covered by the extracted spectra.
- `wdegree`: defines the maximum order of the polynomial fitted to the dispersion relation.
- `wdradius`: defines the search radius for detected lines around predicted positions after the first run of the pattern matching. A larger radius allows to find more lines, but also increases the risk for false or spurious identifications.
- `wreject`: sets the maximum acceptable difference between the calculated wavelength for a line and the value from the line catalog during the fit of the dispersion relation.
- `wmode`: (long slit data only) defines the interpolation mode along the spatial axis to cover missing rows. A value of 0 means no interpolation and is useful to check the spatial coverage of identified lines. A value of 1 will interpolate locally across gaps, while a value of 2 will perform a global fit.
- `wmosmode`: (multi-object spectroscopy data only) defines the interpolation mode along the spatial axis to cover missing rows. A value of 0 means no interpolation and is useful to check the spatial coverage of identified lines. A value of 1 will interpolate locally (within a given slit only) across gaps, while a value of 2 will perform a global fit.
- `dispersion`: provides the first guess of the dispersion in Å/pixel (for binning 1×1) that is used for the pattern matching.

- `peakdetection`: sets the threshold to detect spectral lines. If there are issues with the detection of slits or the number of lines, varying this parameter will often solve them.
- `ignore-lines`: enables the removal of deviating lines from the fit. Please keep in mind that the solution may look better without such lines, but may not be a closer approximation to the real dispersion solution.
- `used_linesets`: For some grisms the arc lamp lines are clustered in certain regions. In order to avoid a fit to the dispersion relation that is dominated by these regions only a subsample of the arc lamp lines is used if `used_linesets` = standard. Setting instead `used_linesets` = standard, extended will include all lines in the catalog.

Below we provide information on how to improve your results by changing the parameters of the **fors_calib** recipe.

4.4.1 Slit Tracing

If you find that the slits in your data were not well traced you may want to change the parameter `cdegree`. Alternatively the slitlets in your data may be too tightly packed. As a default, the **fors_calib** recipe tries to recover untraceable edges by interpolating a global curvature model based on other traceable edges (if they are available). Using this global description of the spectral curvature helps to extract also those spectra whose edges cannot be traced. In some cases however the recipe may find and accept a bad tracing as if it were good, producing a bad global curvature model, and therefore a bad spectral extraction. Setting `cmode` to 0 will suppress the usage of the global curvature model. In this case the recovery strategy of lost spectral edges will consist in replicating the trace of the other available spectral edge (opportunely shifted) of the same slit spectrum. This may improve the results in some cases: however, if a tracing is missing for both edges of a slit spectrum, the spectrum will not be extracted.

4.4.2 Wavelength Calibration

To verify the quality of the wavelength calibration the user should check the displays of the "Graphical Report" and quality plots. While the parameters in the grism tables should generally provide good results some grisms are sensitive to the positions of the slits on the CCD, the line intensities, etc. In these cases it may make sense to try changing certain parameters to improve the results. For problematic cases setting `wmode` (LSS/LSS-like) or `wmosmode` (MOS/MXU) to 0 can be helpful as then only lines that are actually found and detected are used, without interpolation along the spatial axis.

Below we list suggestions to solve problems that may occur during wavelength calibration:

Problems with line detection and identification

Setting `wradius` to 0 will force the pipeline to use only those lines for wavelength calibration that were identified by pattern-matching. This can sometimes help to understand why some lines were not correctly identified.

Faint arc lamp lines

If some of the reference lines listed in the catalog do not peak above a given threshold, they are not used by the pattern-matching task. This can be solved by lowering the value for `peakdetection`. If the faint lines are too close to the noise level you may remove them by specifying their wavelengths in the parameter `ignore_lines`.

Incorrect spectral dispersion

The actual mean spectral dispersion can differ significantly from the default read from the GRISM_TABLE, for instance due to temperature effects or extreme slit positions. In general the pattern-recognition algorithm is quite robust against changes of the spectral dispersion of up to 20%, but for some grisms good results can only be obtained within a very close first-guess. Check for information on **Individual Grisms** below.

Spectra at very large offsets

The observation may include spectra at large offsets along the dispersion axis, so that only part (red or blue) of their full wavelength range is observed. If the line catalog contains too few reference lines in this region (say, less than 5), they might not be enough to define an unambiguous pattern to detect. Setting `used_linesets` to standard,extracted will make additional lines available for a more complete coverage of the bluest/reddest parts of the complete spectral range, which might solve this problem. If there are no extra lines to be used as a reference (as is the case for some grisms), the truncated spectra will be definitely lost.

Valid reference lines are rejected Sometimes the peak detection algorithm may return inaccurate positions of the detected reference arc lamp lines. Outliers are automatically rejected by the fitting algorithm, but if those lines were properly identified, not rejecting their positions may really improve the overall accuracy of the wavelength calibration. In such cases increase the value of the `wreject` parameter, but check the results carefully: a tolerant line identification may provide an apparently good fit, but if this is based on misidentified lines the calibration will include unknown systematic errors.

Systematic trends in the wavelength calibration residuals

This problem may be solved by increasing `wdegree` to 16, but one should be careful to avoid overfitting. This happens preferably at the red and blue ends of the spectra, where the polynomial is so poorly constrained that it also fits the uncertainties of the line positions, incorporating this noise into the solution: the corresponding residuals will be very small, but the calibrated spectra will look very noisy at the edges of the wavelength range. An extreme case of overfitting is, for instance, fitting 4 points with a 3rd degree polynomial: the residuals will be exactly zero, and yet the obtained model will be highly inaccurate. To reduce the influence of noise on the solution one may set `wmosmode` to 1 or 2 (for MOS/MXU data). `wmode` is already set to 2 by default for LSS and LSS-like data.

The calibrated spectra look "noisy" at their ends

This is a consequence of overfitting (see above for details). Here `wdegree` should be reduced. If this introduces systematic residuals one may try to increase `wradius`.

The calibrated LSS spectrum looks distorted

This happens if the global wavelength calibration model is bad, either due to some bad local solutions or due to too few local solutions for global modeling. Rerunning the recipe with `wmode` set to 0 will show the result without any interpolation. The situation can be improved by varying `dispersion` and/or `peakdetection` to maximise the number of obtained local solutions and to get more uniform results in the reduced_lamp_lss.fits image. If the problems persist, setting `wradius` to 0 will limit the lines used to those detected solely by pattern

matching. Once a new set of parameters is found that yields good results set `wmode` again to 2 and rerun the recipe. Also `wradius` may now be set back to its default value of 4.

Individual Grisms

Here we list tips for individual grisms. The grisms are sorted by increasing resolution. Keep in mind that the pipeline does not allow `wdegree` ≥ 5.

- GRIS_150I+OG590 - This grism may need tailored parameter settings for each setup. Increasing `peakdetection` to 300–400 and/or changing `wradius` to higher or lower values may provide better results for certain masks. Changing the `dispersion` within the range 2.9 to 3.8 may be helpful as well.
- GRIS_300V+GG435 - Reducing `wdegree` from 5 to 4 produces in some cases more stable results. In some cases increasing `peakdetection` to 350 can improve the slit tracing.
- GRIS_300I+OG590 - Bad slit tracing can often be improved by increasing `peakdetection` to 400. Changing `wdegree` to higher or lower values can also help.
- GRIS_600B - Setting `wdegree` to 5 sometimes gives better results for both wavelength calibration and slit tracings. Also changing `wradius` to lower or higher values can help.
- GRIS_600RI+GG435 - In some cases the solution can be improved by changing `wdegree` to 3 or 5.
- GRIS_1200B - For this grism it is sometimes impossible to find a solution for all slitlets, especially if the slitlets are widely distributed along the dispersion direction. The user may have to process the data twice, once optimizing for the redder slitlets and once for the bluer ones. In that case varying `dispersion` between 0.32 (blue slits) and 0.38 (red slits) may give good results for extreme slits. The wavelength calibration reacts very sensitively to the presence or absence of the line 4077 Å. If results stay unstable the user should ignore one or more of the following lines via the `ignore_lines` parameter: 3466, 3611, 3965, 4026, 5331, 5401.
- GRIS_1400V - Decreasing wradius to 2 and/or increasing peakdetection to 90 or 120 can help to improve unstable solutions. Some solutions especially for MOS/MXU data can be improved by increasing wdegree to 4.
- GRIS_1028z+OG590 - The solution can be improved in some cases by increasing `wdegree` to 4. Decreasing `wradius` to 2 may improve unstable solutions.

4.4.3 Flat Field Normalization

In general the user will want to remove all traces of the flat field lamp but keep the slit profile (spatial illumination) together with the pixel-to-pixel variation of the detector. The default parameters of the flat field normalization are optimized for this situation (`sradius`, `s_degree`, `d_knots` set to -1, `dradius` set to 10). The spline fit along the dispersion axis `d_knots` ≥ 0 is not well suited for data with very steep slopes due to order separation filters (e.g. GRIS_150I+OG590). Avoiding normalization along the slit (i.e. setting both `sradius` and `s_degree` to -1) keeps the slit illumination and thus allows the user to correct the illumination profile of the science data. This facilitates the fitting of the sky background, which should then be flat along the spatial axis.

4.5 Response Determination

To verify the quality of the response curve determined by the **fors_science** recipe the user should check the corresponding "Graphical report" reduced_std_mos.png of the task **standard_mods**, recipe **fors_science** as in Figure 21.

The reference data for the flux standard stars have entries that identify the presence of strong stellar features (column `STLLR_ABSORP` set to T(rue)). These regions will be ignored during the response fit if `resp_ignore_mode` is set to `stellar_absorption`. Depending on the resolution of the grism such masking may or may not be necessary. If additional regions need to be ignored `resp_ignore_mode` set to `stellar_absorption,command_l` allows the user to specify the additional points (e.g. 4750, 4900) and/or regions (e.g. 8900-9850) with `resp_ignore_po`

In order to avoid the fitting of regions affected by telluric absorption (whose strength may differ between reference spectra and FORS2 data) the static calibration table fors2_telluric_regions.fits contains such regions (per grism, as their width depends on spectral resolution). These can be masked by setting `resp_ignore_mode` to telluric. If additional regions need to be ignored `resp_ignore_mode` set to `telluric,command_line` allows to specify the additional points/regions with `resp_ignore_points`. This is especially useful for data with very strong variations at the edges of the response.

By default `resp_ignore_mode` is set to stellar_absorption,telluric,command_line.

In order to take small scale variations into account when fitting the response the pipeline uses a spline fit with the number of knots defined with `resp_fit_nknots`. For grisms with a small wavelength range and/or only moderate large-scale variation in their response a small number of knots can be sufficient. For grisms with small-scale variations a higher number of knots may be necessary (e.g. fifth demo dataset, see above). One should keep in mind that the maximum number of knots for the spline is the number of unmasked data points – 2 (entering a higher number will cause the pipeline to reduce it to the allowed maximum). Since the knots are distributed at equal distances this means that the distance between two knots is always larger than the distance between two data points. This explains why even at a maximum number of knots the fit may not go through all data points.

4.6 Telluric Correction

The two strong O2 features at about 6870 Å and 7630 Å often show residuals after the telluric correction. This is due to the fact that they are often saturated, which is not noticeable in the low-resolution FORS spectra because the signal is smeared out.

4.7 Science data reduction

1. Sky subtraction

   Sky subtraction `skylocal` and `skyglobal` work on 2-dimensional images, which have been corrected for bias and flat field, but neither wavelength calibrated nor distortion corrected. `skymedian` works on 2-dimensional image which have been wavelength calibrated and distortion corrected.

   - `skyglobal`: activates the sky determination using the curvature of the sky lines to create a supersampled sky spectrum. It does not work well in the presence of significant gradients in flux or

resolution along the slit, but may give good results for ling-slit data.

- `skylocal`: activates the modeling of the sky along each column within a slit in the unmapped data (neither wavelength calibrated nor distortion corrected). It works best for short slits (a few tens of arc seconds) with negligible curvature of sky lines. For long-slit data it therefore should be set to skymedian.
- `skymedian`: determines the sky by taking the median value along the slit for each wavelength column, which is obviously not a good choice in the case of significant gradients in flux or resolution along the slit. `skymedian` is most robust method of the three.
- `cosmics`: performs a cosmic ray rejection on the 2-dimensional frame (only if `skylocal` or `skyglobal` are activated), which is not very robust. Usually the optimum extraction removes all cosmic ray hits. If `cosmics` is used the results should be compared to the results without using `cosmics` to check for artefacts.

2. Spectra extraction

   The values for this section recipe parameters are all provided in unbinned pixels.

   - `slit_margin`: defines the number of pixels to be excluded at the upper and lower slit edge during detection and extraction of objects.
   - `ext_radius`: sets the maximum extraction radius for detected spectra. The default value corresponds to an aperture of 3", which might not be sufficient to cover the spatial extent of an object. For the optimum extractions this parameter can be set to a large value as long as no other spectra are inside the radius.
   - `cont_radius`: sets the minimum distance at which 2 objects of similar brightness do not contaminate each other's spectrum. This parameter is important in crowded regions.
   - `ext_mode`: sets the extraction mode of the 1-dimensional spectra (0 for a straight sum, 1 optimum extraction using the Horne algorithm).
   - `generate_idp` activates the creation of VO/Phase3 compliant FITS tables for each extracted spectrum.

3. Wave Calibration

   - `skyalign`: activates the refinement of the wavelength calibration using sky lines, which is by default switched off (-1), because it should be used only with careful checking. A value greater than -1 gives the order of the polynomial to fit the wavelength offsets.

Also for the science data some useful checks can be made, for which examining them via "Output Files" can be very helpful.

Were all spectra properly wavelength calibrated?

The overall quality of the wavelength calibration can be examined in the MAPPED_ALL_SCI_LSS/MOS/MXU product. It displays the scientific spectra from each slit after removing the optical and spectral distortions. The visible sky lines should all appear perfectly aligned and vertical.

Were all spectra extracted?

For very faint sources or sources with emission lines only the pipeline may not extract the spectra. In some cases changing the sky subtraction from `skylocal=true` and `skyglobal=false` to `skyglobal=true` and `skylocal=false` helps. Alternatively you can try to increase the extraction radius via `ext_radius` or change the masking of the slit edges via `slit_margin`. In the worst case you will have to extract the spectra manually with some other software of your choice.

Spurious objects detected at the slit edges

As a default the **fors_science** recipe excludes objects that are detected within 3 pixels from the slit ends. This might not be enough in some cases and increasing `slit_margin` might help.

Sky subtraction for resolved sources

In case of extended objects filling most or all of the slit, the evaluation of the sky may be strongly biased by the inclusion of signal that actually belongs to the object to extract. Subtracting this contaminated background would actually destroy the object spectrum. For extended objects one should set `skyglobal` to `true` and `skylocal` to `false`. Then the pipeline subtracts a supersampled model of the median sky spectrum observed in all slits from all spectra. This method requires very similar spectral resolution for all slits and will not work well if this is not the case. It is always possible to process the scientific exposures in both ways, one for processing point-like sources and the other for processing spatially resolved sources.

Sky subtraction for curved or tilted slits

Obvious residuals related to the sky subtraction are visible on the extracted slit spectra. In this case one should set `skymedian` to `true` and `skylocal` to `false`. While `skylocal` subtracts the sky before, `skymedian` subtracts after the rectification of the spectral data. The second method performs poorly in comparison to the first, but in the case of curved or slanted slits there is at the moment no other choice than using it.

5 List of workflow tasks

This is the list of all the tasks and associated recipes in the FORS SPECTROSCOPY workflow. Only some of them are needed for scientific reduction, they are indicated by the flag "yes" (triggered by default) or "optional" (triggered only if requested by a workflow parameter). Other tasks are not used for scientific reduction (they are indicated by the flag "no"), they are mainly used for instrument monitoring and they can be executed only by specifying them as target. Note that, when a task is specified as target, all the tasks that generate the calibrations needed for it are automatically executed.

TASK	RECIPE	Used in science reduction	Notes
bias	fors_bias	yes	Computes the master bias frame
calibration_lss	fors_calib	yes	Creates all necessary calibration products to process LSS data
calibration_mos	fors_calib	yes	Creates all necessary calibration products to process MOS data
calibration_mxu	fors_calib	yes	Creates all necessary calibration products to process MXU data
calibration_std	fors_calib	yes	Creates calibration products to process spectrophotometric standard star data
calibration_hc_lss	fors_calib	no	Creates LSS calibration products used for monitoring
calibration_hc_std	fors_calib	no	Creates MOS calibration products used for monitoring
lamp_monitor	fors_sumflux	no	Used for flux monitoring of the lamps
science_spectra	fors_science	yes	Reduces science data
standard_mos	fors_science	yes	Reduces spectrophotometric standard star data
standard_lss	fors_science	yes	Reduces LSS standard star data
telluric_correction	fors_molecfit_correct	optional	Applies molecfit correction to the science data

Table 5.0.0: FORS SPECTROSCOPY pipeline tasks overview

6 Frequently Asked Questions

- **Q1) Where can I find the final reduced data?**

  Answer: all the products of all the datasets and the reductions are saved into the EDPS_data directory, specified when executing the `edps-gui` for the first time. One can decide to export only the final products for selected datasets and only for the desired reduction attempts into another location for further analysis. See Section 2.4 for further instructions.

- **Q2) How do I stop the application?**

  Answer: Proceed as follows:

  1. Press "Stop EDPS" in the Dashboard.
  2. Type Ctrl-C in the terminal where the application is running. If the application doesn't terminate, type Ctrl-C again.
  3. Alternatively, kill the 'panel serve' process on your system, for example:

     ```
     ps -e | grep panel # get the process ID of the gui (<pid>).
     kill -9 <pid>
     ```

- **Q3) I have closed the browser window where the application is running. How can I reopen the application?**

  Answer: Point your browser to: `http://localhost:5006/edps-gui`

- **Q4) Where can I find some data that I can use to test the application?**

  Answer: Install the 'datademo' package provided with the pipeline installation or download the "Demo Data" package from https://www.eso.org/sci/software/pipe_aem_table.html.

  Please note that the demo data can be large (tens of Gigabytes).

  A convenient script to download demo data for any pipeline is also available and can be used from the command line:

  ```
  curl -O https://eso.org/sci/software/apptainer/eso_download_demodata.sh
  bash ./eso_download_demodata.sh
  ```

- **Q5) How can I start the edps-gui if the following message appears?**

  ```
  Cannot start Bokeh server, port 5006 is already in use
  ```

  Answer: The panel server was not closed properly. Kill it by typing:

  ```
  ps -e | grep panel # get the process ID of the gui (<pid>).
  kill -9 <pid>
  ```

- **Q6) How do I get additional support on EDPS or data reduction in general?**

  Answer: For suggestions, questions, or feedback in general, please open a ticket with the EDPS Support team. This https://support.eso.org/new-ticket?ticket%5Bticket_field_13%5D%5Bdata%5D=227 should take you directly to a webpage for creating and `EDPS` feedback ticket, but in case you want to navigate there 'manually', go to https://support.eso.org, login, click on "Submit Helpdesk Ticket", and specify the Help topic: "Post Observations", "ESO Data Processing System [EDPS]".

- **Q7) I have a lot of disk space, but when I install `EDPS` with `pip` or an ESO pipeline with `Homebrew` I get the error message: `Cannot mkdir: No space left on device`. How do I fix it?**

  Answer: This depends on how much disk space is allocated to the `/home`, `/var`, and `/tmp` directories. The final solution would be to resize the space allocated to the in the organization of the filesystem. However, we list here few tricks that might do the job.

  – Clearing the pip .cache to make space for new packages. Type the command:

    ```
    pip cache purge
    ```

    before installing `EDPS`.

  – Redirect the cache, `Homebrew` temporary build directories into a partition with enough space. Set some of the following environmental variables in your `.bashrc` file:

    ```
    export HOMEBREW_CACHE=<path_to_new_cache_directory>
    export XDG_CACHE_HOME=<path_to_new_cache_directory>
    export HOMEBREW_TEMP=<path_to_new_temporary_directory>
    export TMPDIR=<path_to_new_temporary_directory>
    ```

    The first moves only the location of `Homebrew` cache, the second the cache of most applications (instead of the default `/home/username/.cache`), the third moves the directory where `Homebrew` builds, extracts, and saves temporary files (instead of the defaults `/tmp` and `/var/tmp`). The last changes the global system temporary directory and affects most of the linux commands.

  – As extreme measure, one can move the `/home/linuxbrew/.linuxbrew` directory somewhere else, and create a symbolic link in `/home/linuxbrew`. For example:

    ```
    cd /home/linuxbrew
    mv -f .linuxbrew <path_to_new_directory>
    ln -s <path_to_new_directory> .linuxbrew
    ```

    Important note: this operation might break some internal links. Recipes requiring external packages such as `telluriccorr` might not work (impacts on KMOS, XSHOOTER, FORS2, and MOLECFIT pipelines).

6.1 FORS specific questions

1. **Be aware that some of the FORS-SPECTROSCOPY instrumental setups do not have corresponding GLOBAL_DISTORTION_TABLE.**

   You can process the data also without this calibration. In such cases, the spatial distortion of your LSS data is not corrected, which is of importance mainly if you are interested in the spatial information about your targets. The extraction quality of the spectra is generally affected only at the very edges of the field.

The GLOBAL_DISTORTION_TABLEs do not exist for the FORS2 grisms GRIS_200I, GRIS_600V, GRIS_600R nor for any FORS1 grism. It exists for the grisms GRIS_300V and GRIS_300I that have been in use in FORS2 since April 2009.

2. **Some of the slitlets in my science data look badly traced**

   This is most likely caused by an imperfect tracing of the slitlets in the calibration data. Usually an incompletely traced slit can well be seen in normalized master flat field. Because the pipeline detects the slit edges first in the arc lamp frame a probable culprit is the `peakdetection` recipe parameter of the **fors_calib** recipe, which may be too low or too high.

3. **My long-slit spectra show curved sky lines** This is caused by a bad wavelength calibration. Most likely the recipe parameter `peakdetection` in the **fors_calib** recipe was set too low so that spurious lines were detected.

4. **My reduced spectra do not cover the wavelength range I expected**

   Some low-resolution grisms (e.g., GRIS_300V) are subject to second order contamination, even if used with an order separation filter (e.g., GG435). The pipeline will limit the wavelength range of the reduced spectra to the range not affected by the second order contamination (=twice the bluest wavelength present in the observed data).

   Users can adjust the wavelength range following the steps below:

   - (a) If the old response determination was used (recipe **fors_science**).
     One can change the wavelength range in the interactive window of the **calibration_std** task , by editing the `startwavelength` and/or `endwavelength` values or by specifying these parameters on the command line.
   - (b) If the new response determination was used (recipe bf fors_response). The response is computed by using fit-points defined in the table with the tag FIT_POINT_CATALOG, and they cover only the wavelength region unaffected by second order contamination. To extend the wavelength range it is therefore necessary to edit the table before starting the reduction and insert extra fit points according to the needs.

   When extending the wavelength range for the flux standard star (to get flux-calibrated spectra across the extended wavelength range) please note that many flux standard stars are hot stars, for which the second order contamination may distort the response shape. Note also, that if master response curves are used, only the wavelength range without second order contamination is covered.

5. **Why do I see blank (flux=0) regions in the left and right side of my mapped files?**

   The nominal wavelength range used by default by the pipeline has been chosen to allow covering the registered range in the CCD for slits placed both to the very left or to the very right. For high resolution grisms and no filter (Free), the registered wavelength range in the CCD is in fact smaller than the default values, therefore the pipeline pads with 0 values to complete up to nominal range.

6. **Although I am experimenting with different flux standard stars for my observation the flux calibrated spectra do not change. Why?**

   If your dataset contains a master response curve (MASTER_SPECPHOT_TABLE) the pipeline will use that one instead of the response curves derived from individual standard stars. In order to see the effect of the different standard stars you have to remove the master response curve from the input data pool.

7. **There is no master response curve in my datasets - why?**

   There are currently no master response curves for the grisms GRIS_1200g, GRIS_200I+OG590, GRIS_600I+OG590 GRIS_600R+GG435, GRIS_600V+GG435, nor for any spectra observed before January 1, 2015.

8. **There is no flux standard star for my LSS observations - why?**

   The ESO calSelector delivers for LSS data only those flux standard stars that were observed at the same position on the detector as your long-slit spectrum to ensure the same wavelength coverage. In earlier times this rule was not followed strictly and sometimes the standard star was observed at the center of the field-of-view instead. You can look for the additional observations at the ESO Archive instrument specific query forms for FORS2 and FORS1 by setting the following constraints:

   `Start The date of your observations: - 5 days`

   `End The date of your observations: + 5 days`

   `DPR.CATG: CALIB`

   `INS.MODE: MOS`

   `INS.FILT1.NAME:` the order-separating filter of your observations (use `Any` if you had no filter)

   `INS.GRIS1.NAME:` The grism of your observations and selecting for display in addition

   `DP.ID`

   `OBS.TARG.NAME`

   `DPR.TYPE:` to identify the standard star (STD) and its calibrations (FLAT,LAMP and WAVE,LAMP)

   `DET.CHIP1.ID:` to identify the correct chip for mosaic data

   `SEQ.SPEC.TARG:` position at which a MOS standard star has been taken

   For FORS1 data after April 1, 2007 and FORS2 data after April 1, 2002, you will always find 2 files per observations. You need the ones with `DET.CHIP1.ID = CCID20-14-5-3` or `Norma III`.

9. **Part of my flux calibrated spectrum has flux=-1. Why?**

   The response curve may cover a different wavelength region than the science spectrum. Pixels not covered by the response curve are set to -1 in the flux calibrated data.

10. **If I modify the parameters' values, some changes are ignored**

    The FORS pipeline has a special way to handle the parameters which are in the grism tables. Basically, the initial values are always taken from the grism table. Still, during the loop execution, the parameters can be changed to the desired values. If you want to use different values as initial values for those parameters, edit the grism tables (FORS2_GRS_*) to create your own version. The parameters affected are: `dispersion`, `peakdetection`, `wdegree`, `cdegree`, `reference`, `startwavelength`, `endwavelength`, `resp_use_flat_sed`, `resp_fit_degree`, `resp_fit_` and `dradius_aver`.

11. **I am using the maximum number of knots for the spline fit of the response but the fit still does not go through all data points.**

    The maximum number of knots for the spline is the number of unmasked data points – 2 (entering a higher number will cause the pipeline to reduce it to the allowed maximum). Since the knots are distributed at

equal distances this means that the distance between two knots is always larger than the distance between two data points. This explains why even at a maximum number of knots the fit may not go through all data points. A polynomial of very high degree might achieve that, but is rather unstable.

12. **The fit to the response curve shows very strong deviations in the masked regions - how can I avoid that?**

    Some grisms have a very steep slope and/or cover small-scale variations of the detector response so that spline fit with a high number of knots is needed to fit their response. This fit is unconstrained in the masked regions, which can result in very strong deviations. The masking regions have been defined very conservatively. It may well be that the response in the masked regions does not show any significant deviations or only at very few points. In such a case set `resp_ignore_mode` to `command_line` only and add the deviating points at `resp_ignore_points`. Alternatively you can try to exclude the steep part of the response via `resp_ignore_points` and/or use a polymonial fit of lower order, that will not take into account small-scale variations of the response (via `resp_fit_degree` instead of `resp_fit_nknots`).

13. **I used** `use_flat_sed`**=true for my GRIS_1200B data but the flux-calibrated data look very bad.**

    Unfortunately the spectral energy distribution of the flat field lamp use for GRIS_1200B data can vary considerably on short timescales. In such cases the FLAT_SED_<mode> used to correct the effect of the position-dependent response will differ even if they were taken at the same position on the CCD.

    So it will be impossible to correct the effect. If you have LSS data it is very likely that the flux standard star was observed at the same position on the CCD as your science target, so that you may use `use_flat_sed`=false. In the case of MOS/MXU data there is unfortunately no solution for this problem. With `use_flat_sed`=true you can correct the small scale variations of the detector response at the price of introducing spurious flux variations. Setting `use_flat_sed`=false, however, will not correct for the position dependency of the response, which can severely distort the SED of your targets, up to reversing the slope of the spectra (i.e. making a hot spectrum look cool and vice versa).

14. **There are no extracted spectra - why?**

    While this may happen for observations of extremely faint targets the most common case for this behavior are LSS observations from the lower chip, as the main target is usually placed close to the center of the field-of-view, which is on the upper chip.

15. **The coordinates reported by the pipeline for my extracted spectra show offsets - why?**

    There can be several causes for such offsets:

    - (a) Imprecise coordinates for the guide star, which introduce an offset in the world coordinate system recorded in the header of the raw data. This effect causes a rigid shift in right ascension and/or declination.
    - (b) An incorrect value of the reference wavelength recorded in the keyword HIERARCH ESO INS GRIS1 WLEN. The pipeline assumes that this keyword gives the value of the undeviated wavelength for the corresponding grism. An incorrect value will introduce an offset of

$$\texttt{INS.GRIS1.WLEN} - \lambda\_true/step\_size * pixel\_scale \quad (1)$$

An offset 10.5 Å for GRIS_1200B for instance corresponds to 15 steps [0.7Å/px] in wavelength and thus for binning 2×2 and standard collimator to an offset of 15px × 0.25"/px, i.e 3.8", on sky.

Depending on the angle on sky with which you data were taken this offset along the dispersion axis will affect right ascension and/or declination.

- (c) An incorrect wavelength calibration. Errors in the wavelength calibration should be below 1 pixel and thus usually not be noticeable in the derived coordinates. For multi-object spectroscopic data, however, such errors might introduce small shifts between spectra from different slits, which may again affect right ascension and/or declination, depending on the angle on sky of the observations.

16. **The spectra in my rectified LSS data are tilted - why?**

    In rare cases the use of the GLOBAl_DISTORION_TABLE results in tilted instead of horizontal spectra in the MAPPED... data. We are still looking into the causes for this behavior.

17. **Why do my flux-calibrated data not agree with independent photometric measurements?**

    In order to have a true absolute flux calibration several requirements need to be fulfilled:

    - (a) The fraction of the total flux of an object that is contained in the slit depends on the shape of the object, the width and orientation of the slit, and the seeing. Absolute flux calibration requires that all the flux of both the object and the standard star has been collected.
    - (b) The flux that arrives at the telescope depends on the transparency of the sky. Absolute flux calibration requires the same transparency for the observations of the target and the standard star. A change between the flux standard observation and the science object observations of any of the parameters mentioned above will change the flux scale in the final spectrum. To compare the flux calibrated spectrum to other measurements, differences in slit losses and atmospheric conditions have to be taken into account.

    With respect to the faction of flux contained within the slit one should keep in mind that the flux standard stars are observed with a 5" wide slit, while science data are typically observed with slit widths of 0.8" to 1". For a point source a slit width of 0.8"/1.0" used with a seeing of 0.8" means that some 33%/24%, respectively, of the target flux are lost (see also Fig. 17). This results in a too low flux for the flux calibrated spectrum of the target. If the standard star or the target or both are observed under non-photometric conditions (e.g. CLR or THN) their observed flux will be lower than it should be. If the standard star is observed under photometric conditions but the science target is not the flux in the flux calibrated target spectrum will be too low. The opposite happens if the target is observed under photometric conditions but the standard star is not. CLR/THN conditions allow for transparency variations of 10%/20%, respectively.

18. **Does the pipeline combine different detectors chips into a common product?**

    No. The spectroscopic pipeline and EDPS workflow works only on files from the same detector chip. Files from different detectors must not be mixed in processing.

19. **fors_calib - The slit identification fails**

    In case of the MOS/MXU observations with few (usually three or fewer) slitlets the slit identification may fail. Using `slit_ident`=false allows the user to process such data.

20. **fors_calib - The pipeline fails with the error message “The wavelength solution at row <number> does not increase monotonically, which is physically impossible. Try with new parameters”.**

    Non-monotonic dispersion relations are often due to spurious detections. In such cases try to decrease `wdegree` and/or increase `peakdetection`. A further possibility is to decrease `wradius`.

21. **fors_calib - The pipeline fails for wide slit data**

    If very wide slits are used, the reference lines become accordingly wider (and display a box-like, flat-top profile). The calibration recipe can handle this in case of well isolated lines, but if nearby lines blend together it is impossible to safely determine their positions. There is no solution for such data.

22. **fors_calib - The pipeline fails with error "No slits could be detected".**

    The recipe cannot detect the slit edges with default value of `peakdetection`, stored in the GRISM_TABLE. In such cases try to change `peakdetection`.

23. **fors_science (Response) - What can I do if I get a problem with fors_calib recipe stating that there is no good wavelength solution for a list?**

    This is a side effect of a wrong slit position determination. In fact, this problem occurs during the refinement of the slit positions if slit identification has been performed. The problem will likely be addressed in future releases. For a workaround in the meantime, try running the pipeline with parameter `slit_ident` set to false.