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

XSHOOTER EDPS-GUI tutorial

1 rev. 20260516

Issue 1

Date May 16 2026

Prepared:	Michael Marsset	May 16 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	15/05/2026	all	First public release for 3.8.3

This page was intentionally left blank

Contents

1 Introduction

1.1 Scope

This document describes how to reduce XSHOOTER data with the `edps-gui` (Graphic User Interface), the dashboard of the ESO Data Processing System (`EDPS`), which the recommended interface to reduce data from ESO telescopes. Details on the XSHOOTER 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 XSHOOTER pipeline itself, consult the pipeline manual available at: https://www.eso.org/sci/software/pipe_aem_table.html.

Note: this tutorial refers to:

- XSHOOTER instrument pipeline named `xshooter`, version 3.8.3.
- XSHOOTER workflow: `xshooter.xshooter_wkf`
- `EDPS` version 1.5.7.
- `edps-gui` version 1.0.

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 consist 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. This workflow can the 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 an 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 processes 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 (definying 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 path 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 XSHOOTER demo data. We assume that the `EDPS`, `edps-gui`, the XSHOOTER 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/software/pipe_aem_main.html.

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).

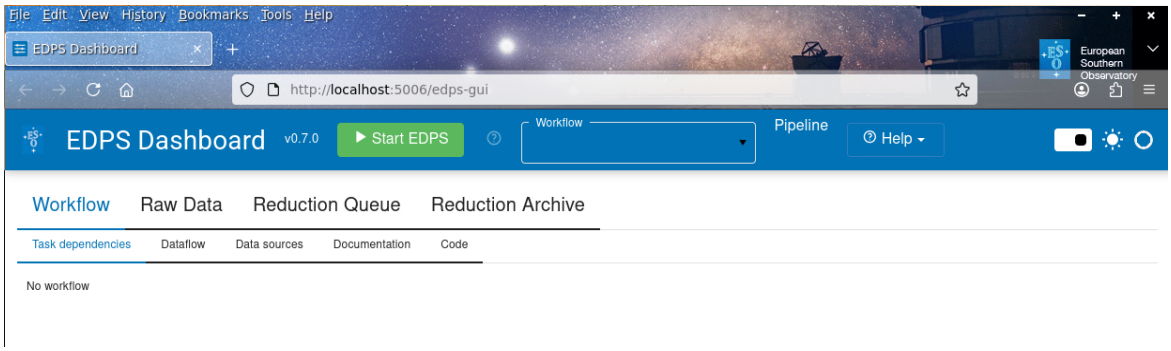

Figure 1: The empty `edps-gui` Dashboard; the underlying `EDPS` engine has not yet been started and no workflow has been loaded.

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

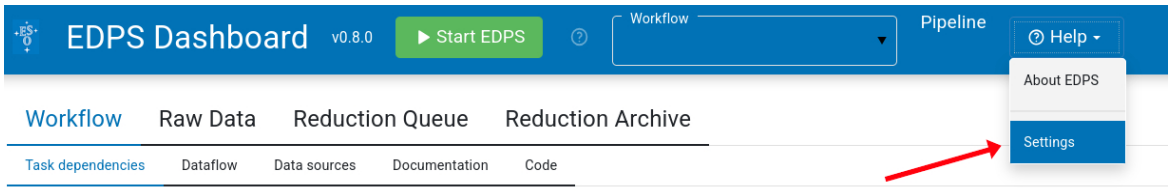

Figure 2: The “Help” –> “Settings” menu.

4. On the browser window with the dashboard, press the button ‘Start EDPS‘.
5. Choose the `xshooter.xshooter_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.

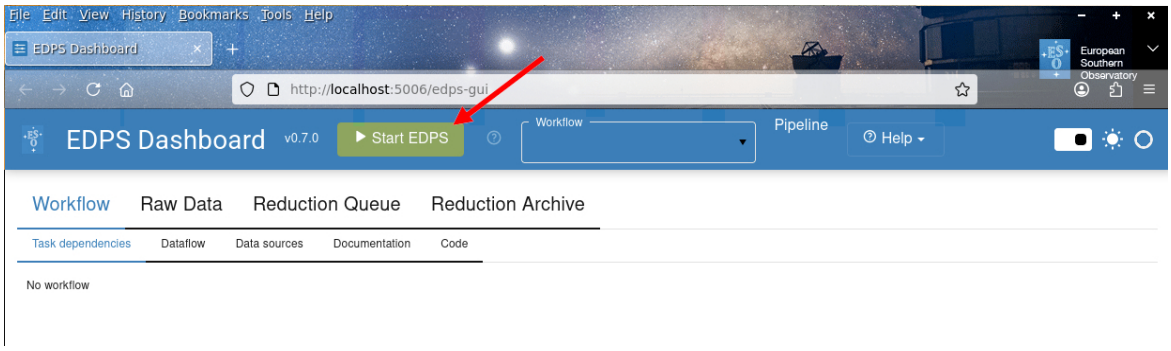

Figure 3: The “Start EDPS” button.

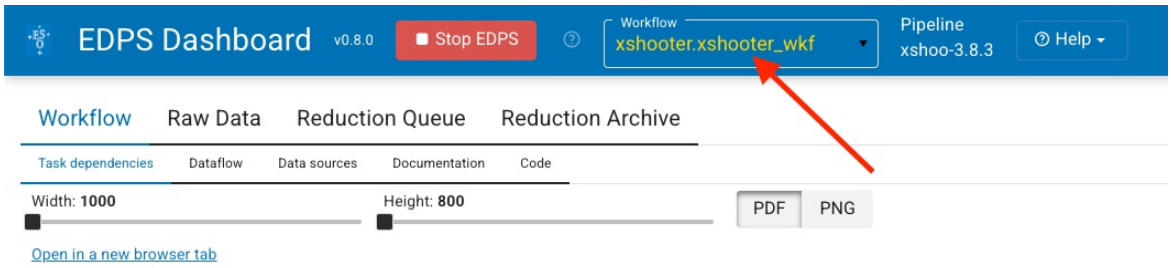

Figure 4: The `edps-gui` with the XSHOOTER workflow loaded.

2.2 Selecting the input data

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

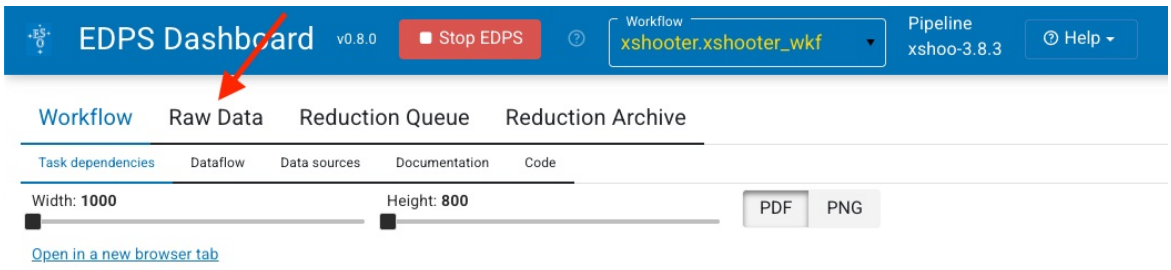

Figure 5: How to select RAW data Tab.

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

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.

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

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

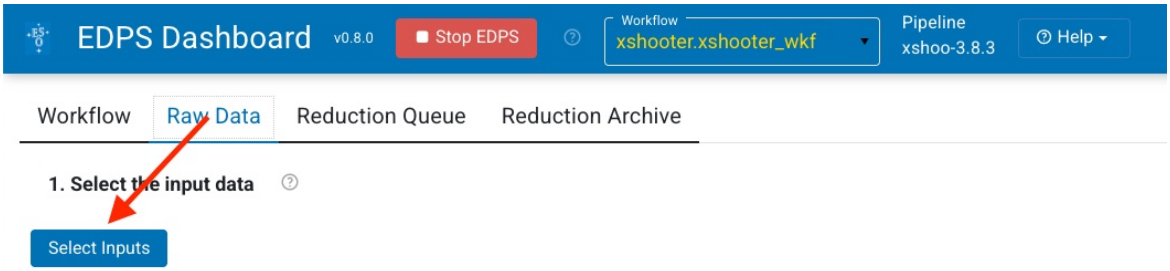

Figure 6: How to select input data.

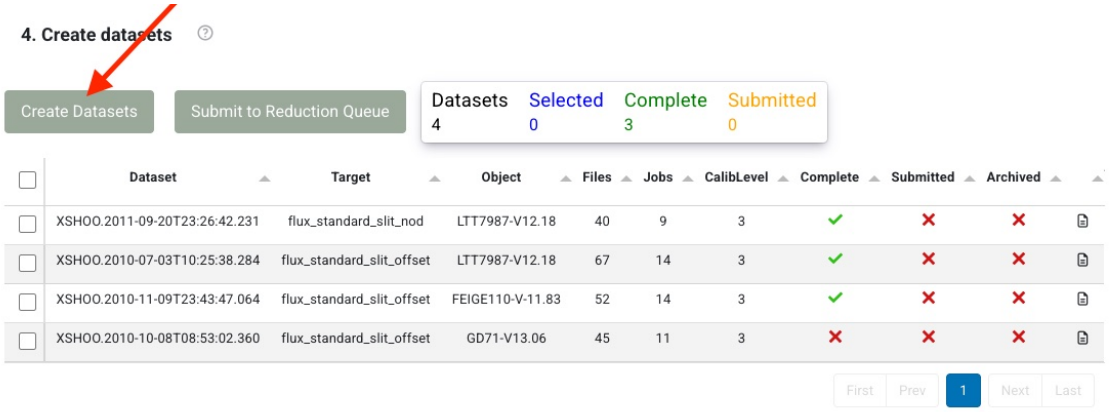

Figure 7: How to inspect the input data directory to create datasets.

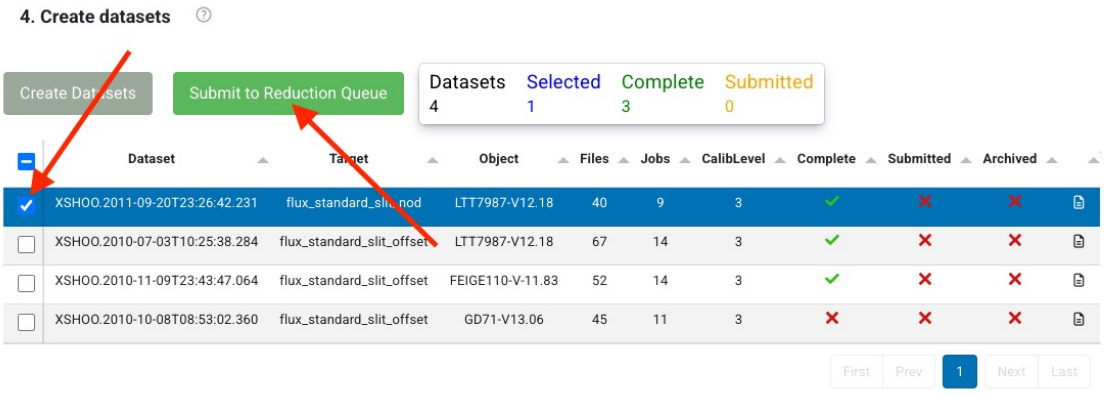

Figure 8: How to send the selected datasets to the Reduction Queue for processing.

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

2.3 Start the reduction

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

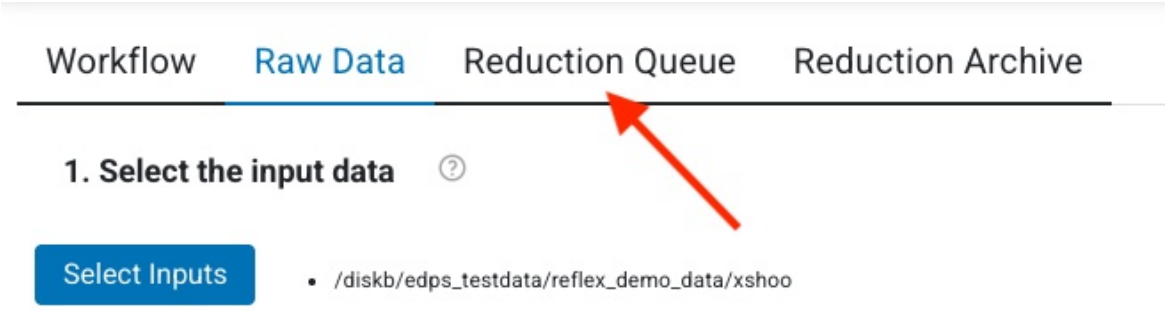

Figure 9: How to select Reduction Queue tab.

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.

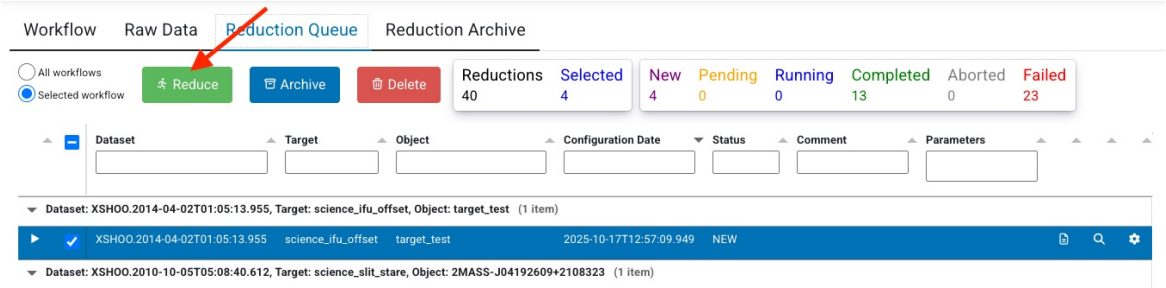

Figure 10: Reduce.

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

Almost all processing tasks can display the input raw frames and the products in the so called "quality plots", which can be inspected from the ‘Reduction Queue‘ window. Those associated for the main product can be inspected by pressing the magnifying glass symbol at the right side of each dataset. To inspect those associated to each individual job (if created),

- 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. Only plots for completed jobs can be inspected.

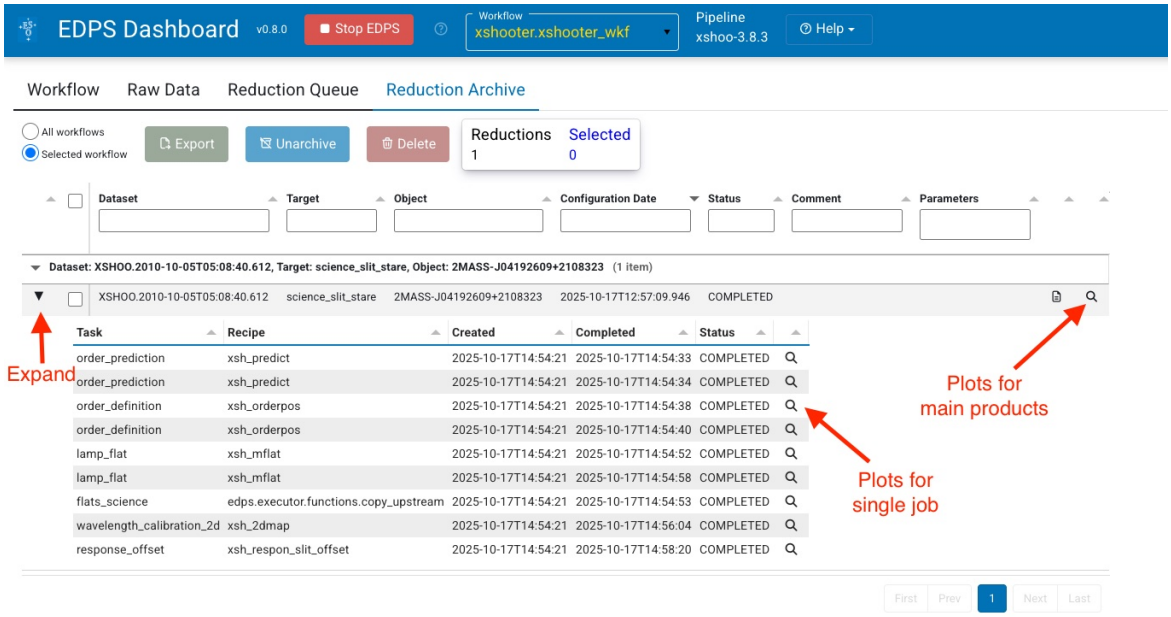

Figure 11: How to look for quality plots from the Reduction Queue tab.

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. Additionaly, even if all products for all tasks are saved in the EDPS_data directory, the most important producs 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.
2. Go in the Reduction Archive tab and click on the 'Export' button. A new tab window appear where you can indicate the directory you want to copy your final products; finally press "Export" to copy the data.

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:

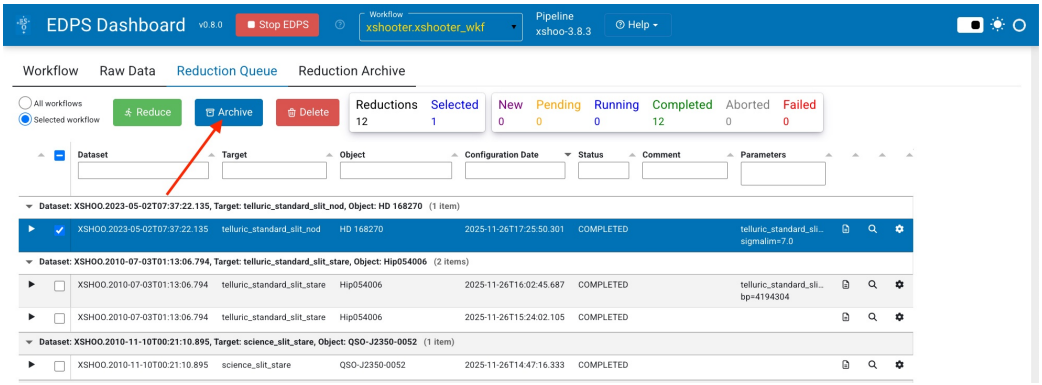

Figure 12: How to archive a completed reduction from the Reduction Queue tab.

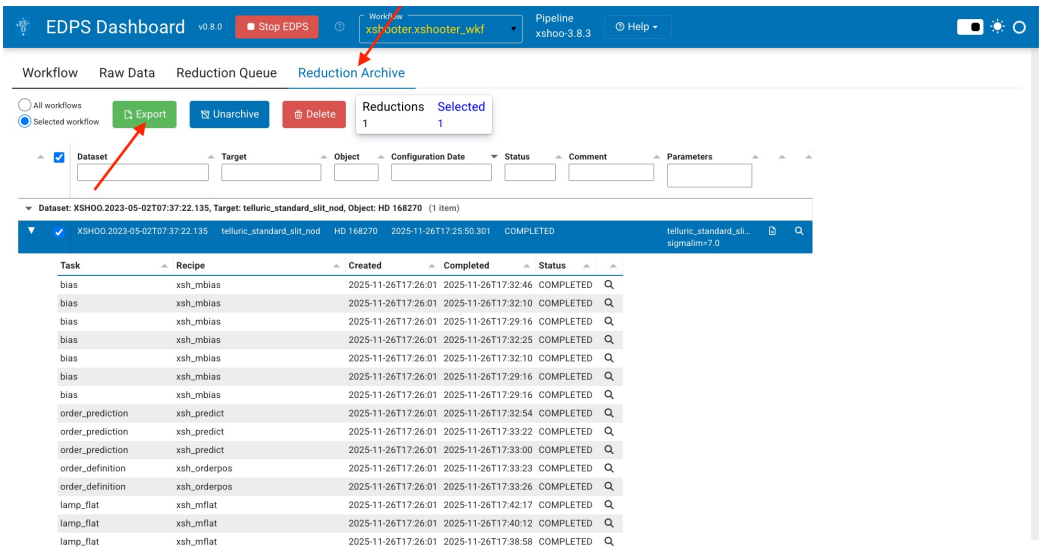

Figure 13: The reduction archive tab. This table contains all the different configurations of datasets that are declared "finished" and removed from the Reduction Queue. From this page, the user can export the most important files into a desired local directory.

- The 1D spectrum, telluric corrected, obtained from the combination of individual exposures. Its name format is: SPECTRUM_COMBINED_<target_name>, where target name is read from the header.
- The individual 1D spectra prior to combination (either telluric corrected or not). They are in a subdirectory of the combined spectrum they refer to their name formats are SPECTRUM_<arm>_<arcfile> or SPECTRUM_TELLCORR_<arm>_<arcfile> if telluric corrected. The values of arm (UVB, VIS, NIR) and arcfile (the file name of the raw science exposure) are read from the file header.
- The datacubes obtained from individual exposures for IFU observations. Its name format is: DATACUBE_<arm>_<arcfile>.

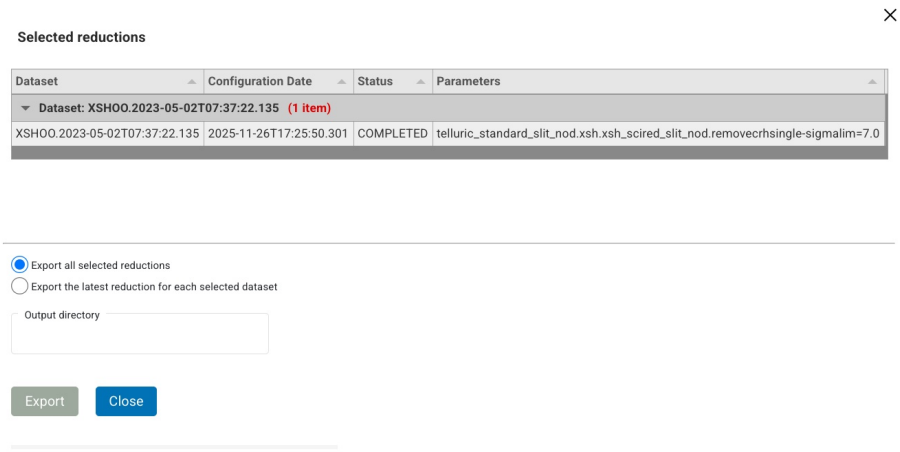

Figure 14: The EXPORT dialogue window, where the user can decide which reduced configuration to save and where.

3 The XSHOOTER data reduction flow.

The overall data flow of the XSHOOTER 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 `xshooter.xshooter_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 `xshooter.xshooter_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

3.1 Generate Master Bias

This step is carried in the task **bias**, which runs the recipe `xsh_mbias`.

Produces a master bias for UVB/VIS arms. NIR frames do not use bias frames.

Customization

Recipe parameters:

- Choose stacking method (average, median) with `stack-method`.
- Adjust sigma-clipping with `klow` and `khigh`.

3.2 Generate Master Dark

This step is carried in the task **dark**, which runs the recipe `xsh_mdark`.

Produces a master dark. Optional for UVB/VIS (dark current negligible). For NIR it is only needed for stare observations; generally not used in nodding/offset where ON–OFF subtraction removes the dark.

Customization

Recipe parameters:

- Enable the use of darks for UVB/VIS data reduction with `use_optical_dark`.
- Change `stack-method` and `klow` / `khigh` thresholds to adjust stacking behaviour, e.g. for better cosmic-ray rejection.

3.3 Fit Orders

Computes initial guesses for the wavelength solution and order positions. Performed in two steps, described hereafter.

3.3.1 Instrument Model Prediction

Recipe: `xsh_predict`

Uses the instrument physical model and information about ambient conditions during the observations (e.g. atmospheric pressure, temperature, instrument setting) to predict line positions and a first dispersion solution.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode. See `configure_reduction.md` for more information.

3.3.2 Order Tracing (Determining Order Geometry)

Recipe: `xsh_orderpos`

Uses pinhole order-definition frames (ORDERDEF_*) to trace the location and curvature of each echelle order on the detector. Produces order tables that later help rectification and wavelength calibration.

Customization

Recipe parameters:

- Tuning of detection thresholds can help when continuum levels are low or orders are partially vignetted.

3.4 Generate Master Flat

Recipe: `xsh_mflat`

Produces a master flat, refined version of the table from `xsh_orderpos` and a bad pixel map for each arm. The flat defines the detector response and updates the true geometry of the orders from slit illumination. For IFU flat fields the edges of the slices are traced.

Customization

Recipe parameters:

- Adjust bad-pixel handling (via `decode-bp`) if master flats saturate or raise quality control errors.
- Change `stack-method` and `klow` / `khigh` thresholds to adjust stacking behaviour, e.g. for better cosmic-ray rejection.

3.5 Wavelength Calibration

The wavelength calibration creates the wavelength and spatial resampling solutions and computes the arc-line tilts and instrumental resolution. It is done in two steps:

3.5.1 2D Mapping

Recipe: `xsh_2dmap`

Determines the two-dimensional wavelength solution needed to resample the orders. Creates `WAVE_TAB_2D_ARM`, which provides a full mapping from detector coordinates to wavelength+slit coordinates, and `SPECTRAL_FORMAT_TAB_`, which defines, for each order, the wavelength range, pixel boundaries, predicted order edge traces, and spatial-to-spectral coordinate conversion.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode.

Recipe parameters:

- Adjust line-detection thresholds for weak arcs via `detectarclines-fit-win-hsize` and `detectarcline`. These parameters have to be small enough not to include a doublet but large enough to be able to detect and fit the line.

3.5.2 Wavelength Calibration

Recipe: `xsh_wavecal`

Computes arc lines tilt and resolving power.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode.

3.6 Flexure Compensation

Recipe: `xsh_flexcomp`

Refines the wavelength solution to account for flexure, especially when arcs are not taken at the same rotator angle as science.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode.

3.7 Flat Strategy

This allows the user to choose between two strategies to select a flat field for the flux calibrator. The default strategy is to use the same flat as for the science observation. The alternative is to use the flat field selected by the rules, i.e., those taken closest in time to the flux calibrator.

Customization

Workflow parameters:

- By default, the `use_flat` parameter is set to `science`, meaning the flats used for science frames are also applied to standard stars. Set it to `standard` to use the flats taken closest in time to the standard-star observations.

3.8 Instrument Response and Efficiency

Recipes: `xsh_respon_slit_stare`, `xsh_respon_slit_offset` and `xsh_respon_slit_nod`

Use standard stars to derive the per-order and merged instrument response (mapping detector counts to physical flux), the blaze correction and the telescope + instrument + detector efficiency. Used for flux calibration of science exposures. The pipeline does not create response curves for IFU data, and they are therefore not flux-calibrated.

3.9 Science Reduction SLIT

Recipes: `xsh_scired_slit_stare`, `xsh_scired_slit_offset` and `xsh_scired_slit_nod`

Perform the data reduction:

- Prepare the science frame (bias/dark/inter-order background correction, flat-fielding).
- Rectify orders using the model and wavelength solution.
- Perform sky subtraction (mode-dependent).
- Localize the object and extract `ORDER1D` and `MERGE1D` products.

When several exposures are fed together, the recipes stack them (mean/median) before extraction, and you get one combined 2D and 1D spectrum per run, not one per exposure. To obtain one spectrum per exposure (or per AB pair in nodding), you must run `xsh_scired_slit_*` separately on each exposure or nod pair.

Customization

Workflow parameters:

- By default, `telluric_correction_mode=standard` derives the atmospheric parameters from the telluric standard. Set it to `science` to derive them directly from the science frame, or to `none` to disable telluric correction.
- Select between physical-model (recommended) and polynomial mode.

Recipe parameters:

- Change `stack-method` and `klow` / `khigh` thresholds to adjust stacking behaviour.
- Sky modelling: Use `sky-method` to select the method. In the NIR, `BSPLINE2` provides the best residuals, `BSPLINE1` is faster and acceptable for UVB/VIS, and `MEDIAN` is the fatest but may leave residuals.
- Sky region selection: use parameters `sky-position1`, `sky-hheight1`, `sky-position2`, and `sky-hheight2` to manually select sky zones. Required when object not centered, multiple objects on slit, or strong NIR gradient.

- Spectroscopic extraction: Standard extraction (`localize-method=MANUAL`) recommended for faint sources. Automatic detection (`localize-method=AUTO`) is usually fine for bright sources.

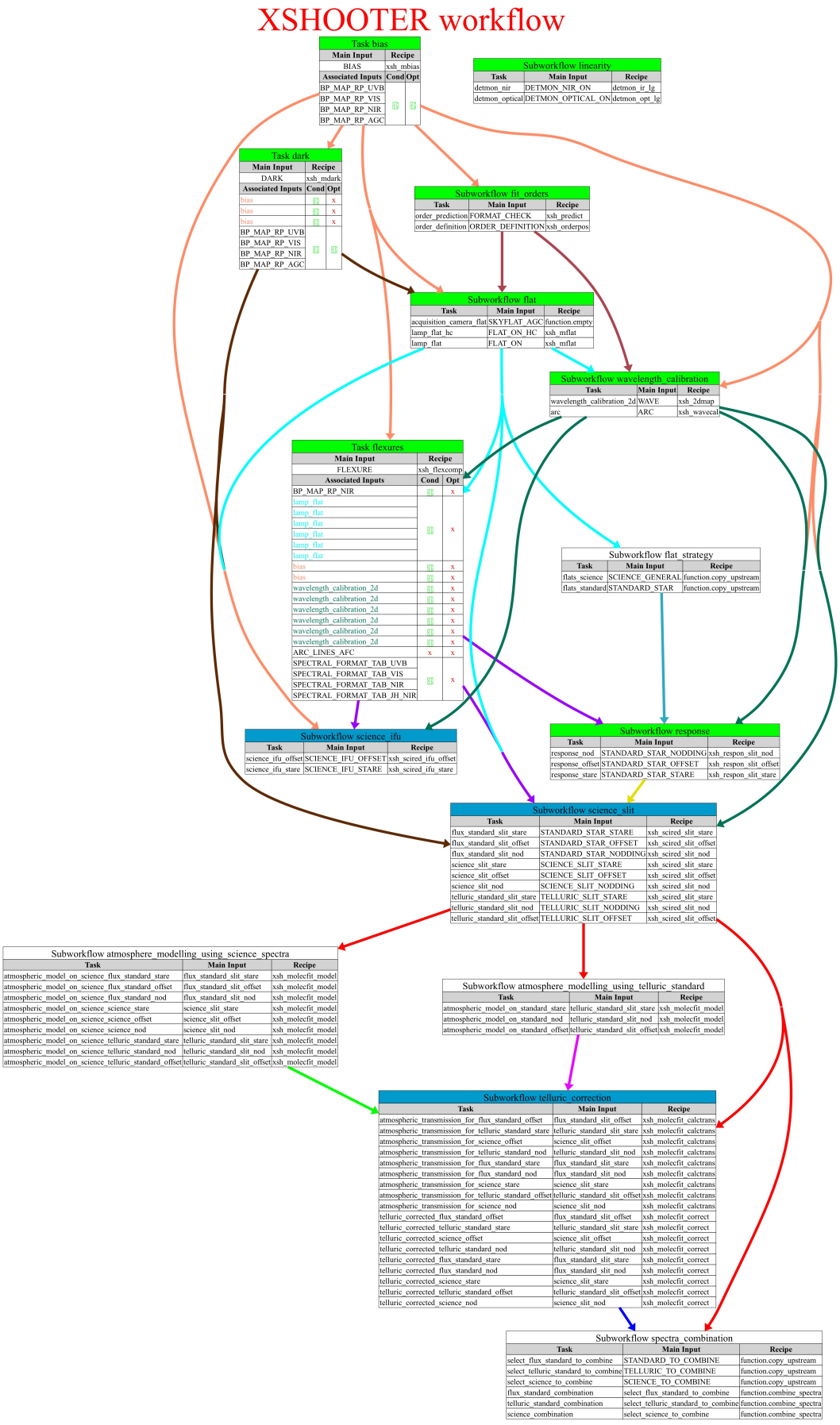

Figure 15: The data reduction cascade of the XSHOOTER workflow.

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 og 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 16.

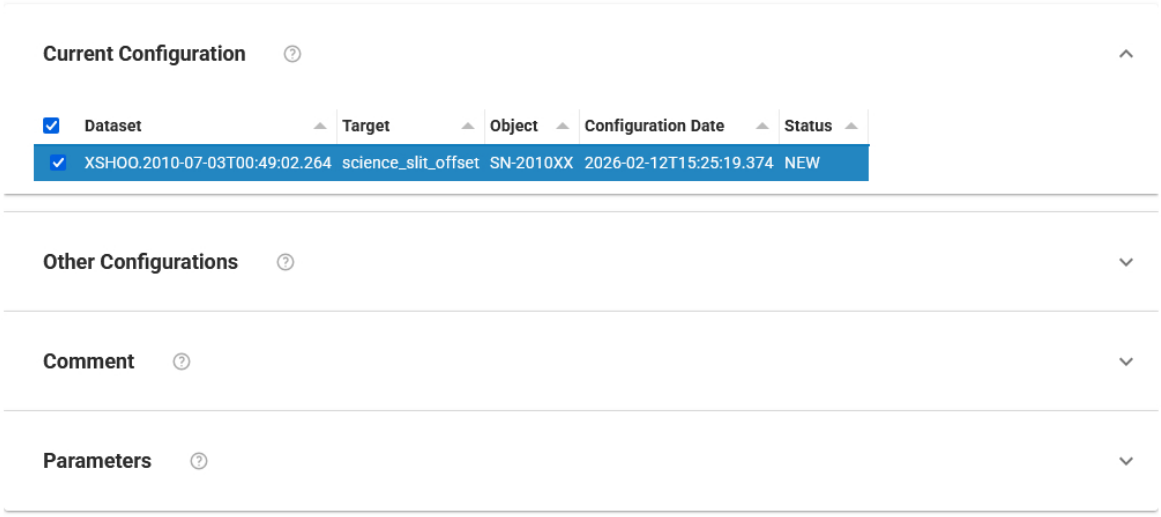

Figure 16: The Reduction Configuration editor. It contains 4 sections, that indicate the current configuration, list of other configurations to set, comments to insert, and the parameters to modify.

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 17).
- **Other configurations.** It allows to specify other configurations, to which the changes shall be copied to (Figure 18.
- **Comment** It allows to specify a comment to describe the configuration. It is possible to append or replace a comment (Figure 19). 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 20 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.

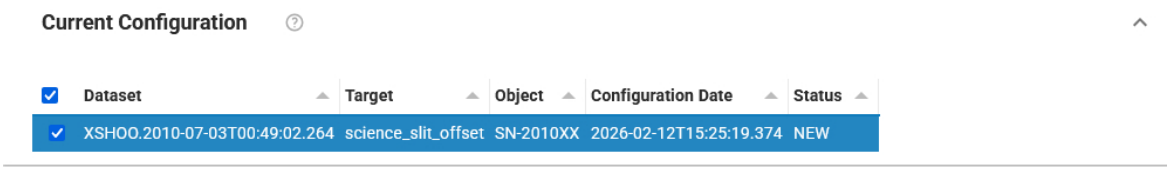

Figure 17: The first part of the Reduction Configuration Editor, that indicates the selected configuration.

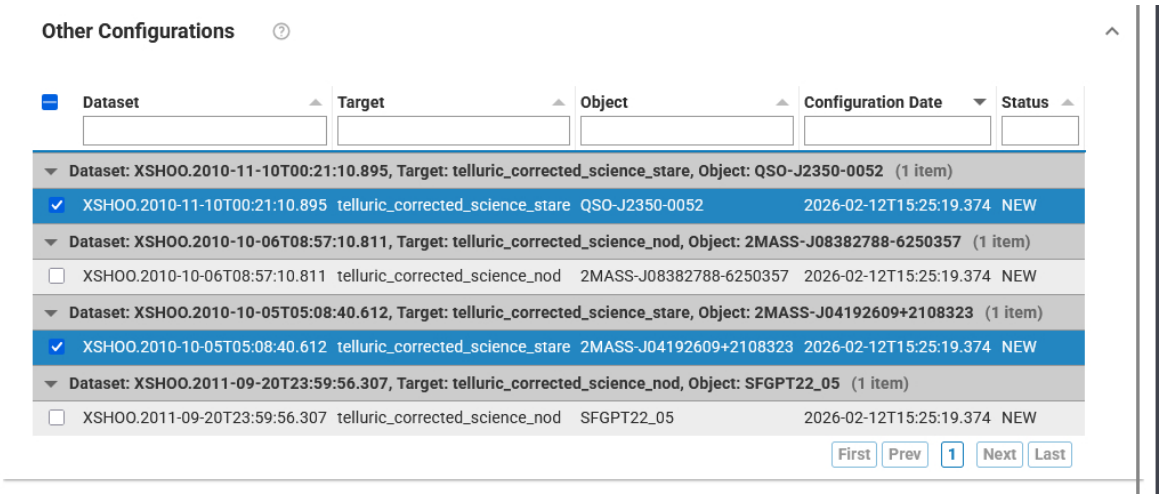

Figure 18: The second part of the Reduction Configuration Editor, that indicates other configurations for which we'd like to apply the changes.

For XSHOOTER the following workflow parameters can be adjusted when the `Target Category` is set to `science`:

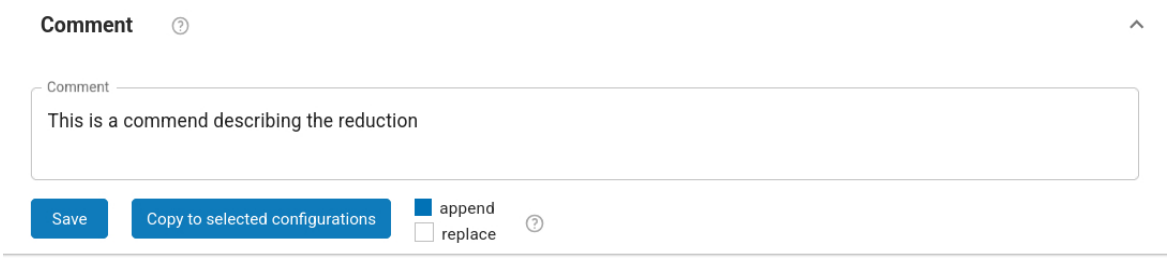

Figure 19: The third part of the Reduction Configuration Editor, that allows to specify a comment to the selected configurations.

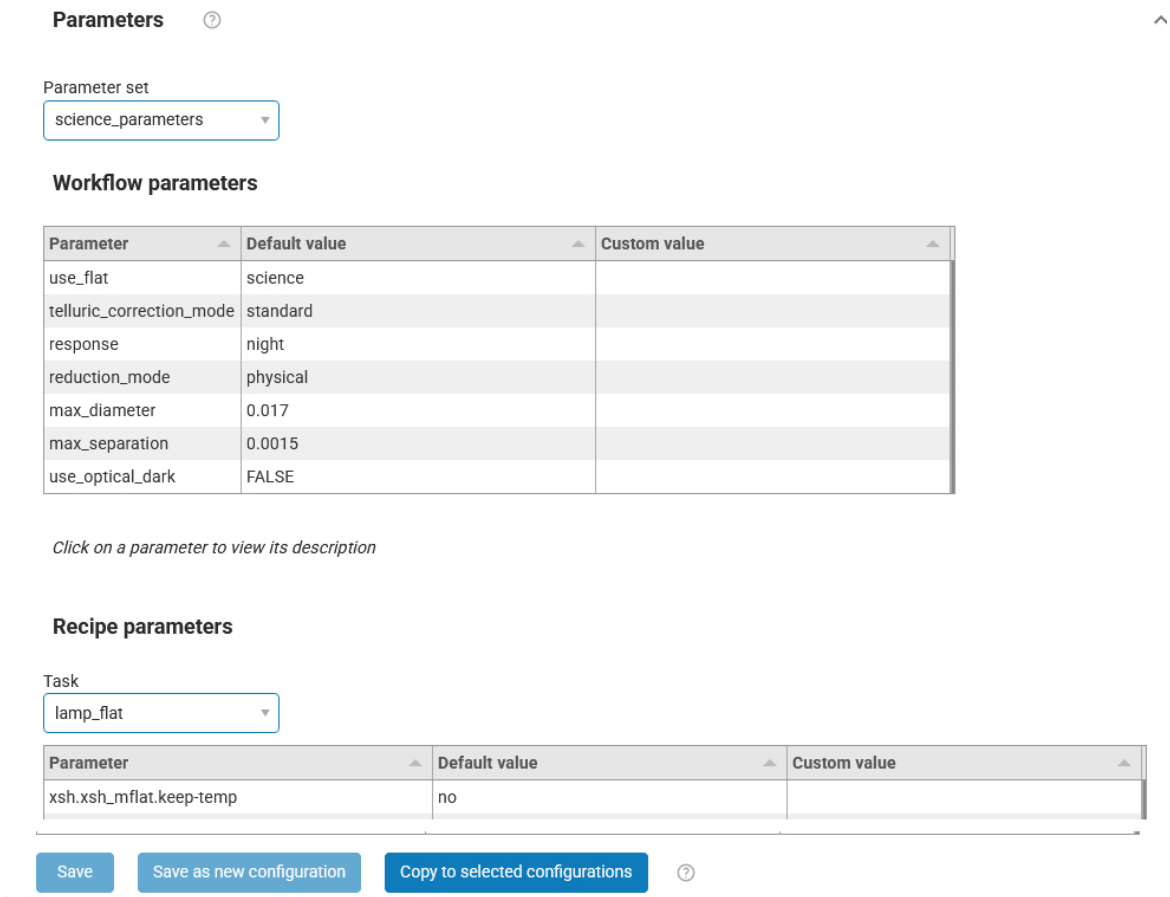

Figure 20: The fourth part of the Reduction Configuration Editor, that allows to specify the parameters sets and the recipe parameter per task. These settings can be applied to the "Selected Configuration" (Fig. 17) or to the "Other Configurations (Fig. 18).

- `use_flat`: By default, this parameter is set to `science`, meaning the flats used for science frames are also applied to standard stars. Set it to `standard` to use the flats taken closest in time to the standard-star observations instead.

- `telluric_correction_mode`: Atmospheric parameters can be derived either from a telluric standard or from the science frame itself. Set this parameter to `standard` (default) to use a telluric star observed the same night with the same instrument setup. Set it to `science` to derive the parameters directly from the science. Use `none` to disable telluric correction.
- `response`: This parameter selects the response curve used for flux calibration. `night` (default) uses the response from the standard star observed that night; if unavailable, fall back to the master response. `master` uses the master response from CalSelector, built by combining standards from multiple nights.
- `reduction-mode`: Select between the physical-model mode (recommended) and the polynomial mode to map each wavelength onto the CCDs. The physical model derives the solution from the actual optical path, while the polynomial method uses empirical multi-coefficient fits per order. If the reduction fails or the physical model appears inconsistent, the polynomial mode can be used for troubleshooting or as a fallback.
- `max_diameter`: TODO
- `max_separation`: TODO
- `use_optical_dark`: This parameter controls the use of a dark frame in the UVB/VIS data reduction. Because the dark current is negligible at these wavelengths, it is set to `FALSE` by default. Set it to `TRUE` if you wish to enable dark correction.

4.3 Troubleshooting

This section provides guidance for diagnosing and resolving common issues encountered during the XSHOOTER data-reduction cascade.

4.3.1 Not enough arc lines detected

The recipe `xsh_2dmap` may fail with an error indicating that not enough arc lines were detected. Typical expected values for `QC.NLINE.FOUND.CLEAN` are:

- UVB: $\sim$ 2550 lines
- VIS: $\sim$ 3950 lines
- NIR: $\sim$ 1300 lines

Verify the quality and consistency of the input calibration products, in particular:

- `ARC_LINE_LIST_ARM`
- `THEO_TAB_MULT_ARM` (poly mode)
- `ORDER_TAB_EDGES_SLIT_ARM`

- `WAVE_TAB_GUESS_ARM`

In addition, adjust the line-detection parameters:

- increase `detectarclines-search-winhsize`
- decrease `detectarclines-min-sn`

to ensure more arc lines are detected.

4.3.2 Sky subtraction residuals (especially NIR)

Residual tilts between the sky model and the observed 2D frame can introduce sky–subtraction artefacts, particularly in the NIR. In practice, this manifests as alternating under- and over-subtracted regions around OH emission lines.

Possible workarounds:

- Verify the accuracy of the `xsh_2dmap` products. Run the reduction chain in physical-model mode, which provides a more robust 2D geometry than the polynomial mode. Ensure that the residuals after model optimisation are small; if needed, re-run `xsh_2dmap` with additional iterations.
- Try setting `sky-method` = `MEDIAN`. This can improve stability when the BSPLINE models fail due to slight geometric mismatches.
- Apply flexure corrections computed by `xsh_flexcomp`. Flexure compensation can realign the sky model and reduce systematic sky–subtraction residuals.

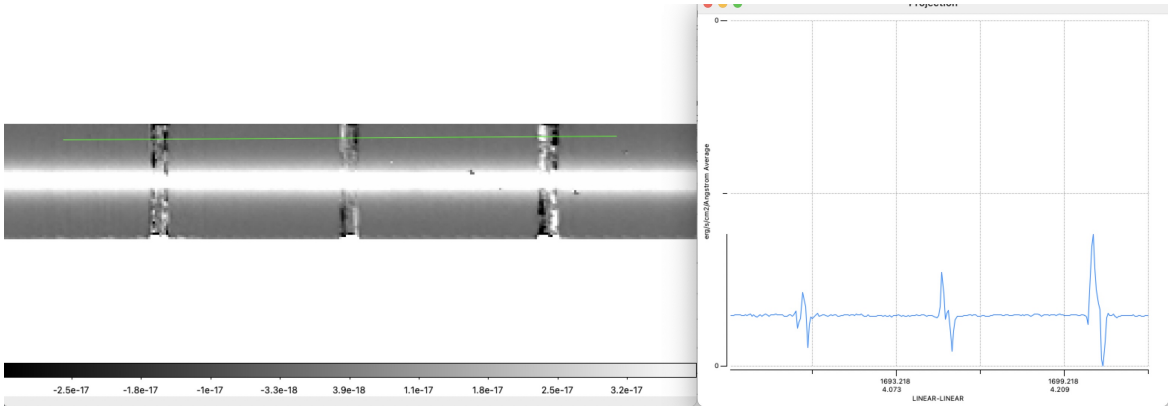

Figure 21: Example of alternating under- and over-subtracted regions around OH emission lines, caused by a slight tilt of the sky lines relative to the object trace. The plot on the right shows the flux profile extracted along the green line in the 2D image.

4.3.3 Trace-localization failures

`xsh_scired_slit_XXX` may fail when `extraction-method` = `LOCALIZATION` and `localize-method` is set to either `GAUSSIAN` or `MAXIMUM`.

This issue typically occurs with low S/N science exposures, where the object trace cannot be reliably detected using either the Gaussian cross-order profile or the maximum-detection algorithm.

The user should try adjusting `localize-slit-position` and `localize-slit-hheight` to guide the trace-localization process, or alternatively set `extraction-method` = `MANUAL`, which avoids automatic trace detection altogether.

4.3.4 Two object traces in the slit

Reduce each object trace separately by providing appropriate values in the `xsh_scired_slit_XXX` recipe for the parameters `sky-position1`, `sky-hheight1`, `sky-position2`, and `sky-hheight2`. These parameters define the specific regions of the slit used to estimate the sky during single-frame sky subtraction.

By default, all four parameters are set to zero, meaning that the sky is taken from all pixels outside the object localization region (and outside the masked slit edges). However, when multiple objects are present along the slit, or when the default choice is not appropriate, the user should manually adjust these positions.

Both the central positions and the half-heights of the sky regions are expressed in arcseconds.

4.3.5 Order-edge artefacts in extracted spectra

Artefacts may affect the edges of the 2D and 1D orders, degrading the quality of the merged extracted 2D and 1D spectra.

In such case, use the appropriate reference format-check frames and verify that the values of `WLMIN` and `WLMAX` in `xsh_scired_slit_XXX` are correct. They should correspond to the last wavelength imaged on the illuminated part of the order, divided by 1.007.

4.3.6 Small jumps between orders

Small discontinuities may appear between orders during merging, typically caused by slit-edge artefacts introduced by the flat-field. To mitigate this, we recommend setting `extract-method` = `LOCALIZATION` in `xsh_scired_slit_XXX`.

4.3.7 Overestimated uncertainties in the extracted spectrum

This issue is typically caused by slit-edge artefacts introduced by the flat-field. To mitigate it, set `extract-method` = `LOCALIZATION` in `xsh_scired_slit_XXX`.

4.3.8 The extracted merged 1D spectrum looks poor

The `xsh_respon_slit_xxx` and/or `xsh_scired_slit_xxx` recipes may have failed to localize the trace correctly during extraction.

4.3.9 Spectral regions of zero flux values

Large regions of zero-flux values in the final extracted 1D spectrum may be caused by spurious Cosmic-Ray (CR) detections.

CR-contaminated pixels are identified using the van Dokkum algorithm (2001, PASP, 113, 1420) applied to the input science frames.

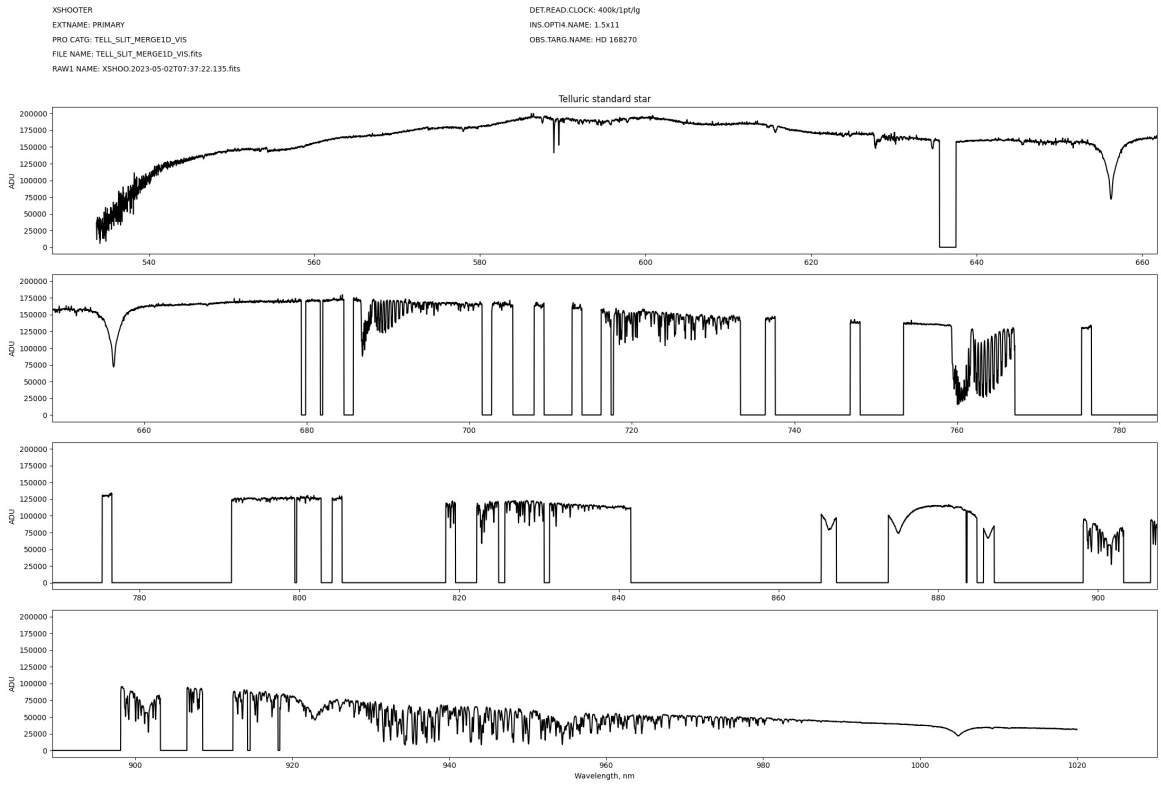

Figure 22: Here we show the VIS spectrum of the telluric standard star Hip089684, as displayed in the `Graphical reports`. The observation was obtained on May 1, 2023 (OB identifier: `XSHOO.2023-05-02T07:37:22.135`) under excellent seeing conditions (0.4”). With the default value `removecrhsingle-sigmalim`=5.0, the algorithm produces spurious cosmic-ray detections, which in turn lead to extended wavelength intervals with zero flux in the extracted 1D spectrum.

4.3.10 IFU cube not well aligned

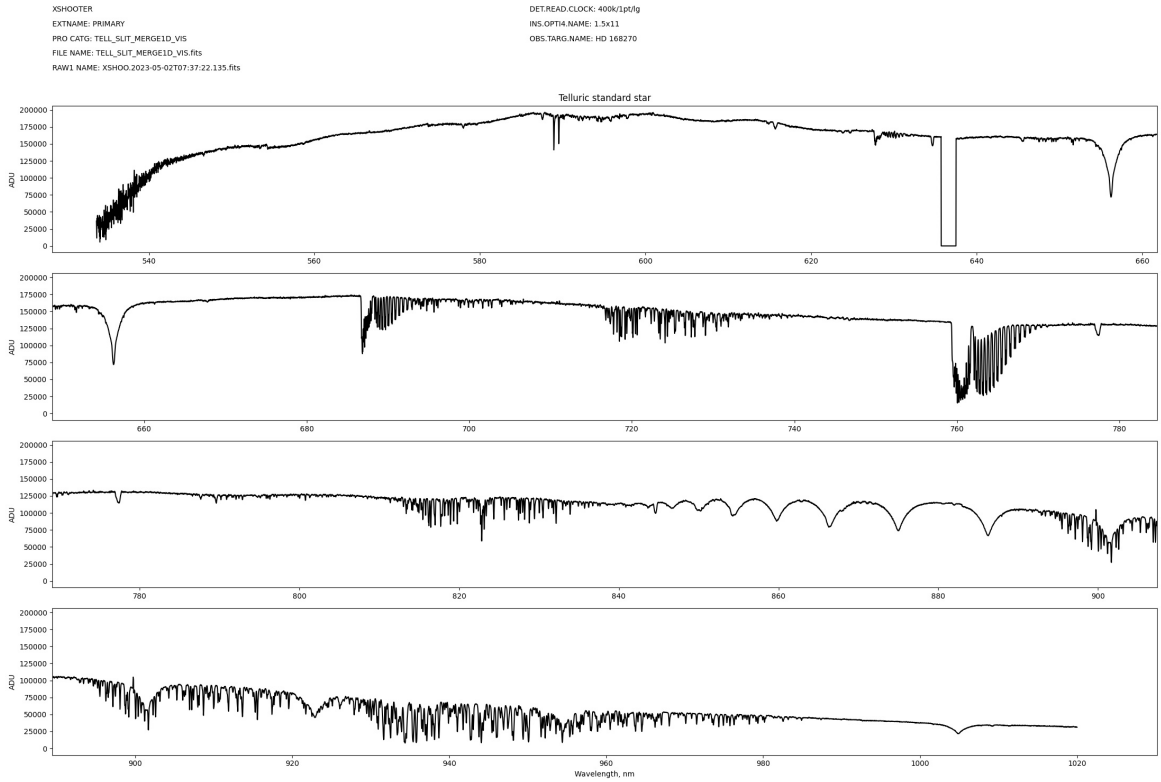

Figure 23: Same as before, but with `removecrhsingle-sigmalim`=7.0. In this case, this avoids the flagging of good pixels as CR-affected.

5 List of workflow tasks

This is the list of all the tasks and associated recipes in the XSHOOTER 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
task1	recipe1	yes	comment 1.
task2	recipe2	no	comment 2.
task3	recipe3	optional	comment 3.

Table 5.0.0: XSHOOTER 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).