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

X-Shooter Pipeline User Manual

VLT-MAN-ESO-14650-4840

Issue 12.0

Date 2013-04-29

Prepared:	Andrea Modigliani, Daniel Bramich		
	Name	Date	Signature
Approved:	J. Vernet		
	Name	Date	Signature
Released:	P. Ballester		
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0.0	27/05/2010	all	public release
1.1.0	21/06/2010	sec 10,11	more illustrations
1.2.0	29/10/2010	all	release update, more illustrations, test scripts description, fix typos
4.0	30/11/2010	all	fix typos
5.0	03/05/2011	all	Release update
6.0	07/07/2011	all	Release update
7.0	05/04/2012	all	Reflex support release
8.0	23/04/2012	all	Reflex update
9.0	15/05/2012	typos	Release update
10.0	22/08/2012	typos	Release update
12.0	29/04/2013	all	Science grade reflex release

Co-authors

Name	Institute	What
Daniel Bramich	ESO	User Manual review
Paul Bristow	ESO	Physical model functionality
Frederic Royer	Galaxies, Etoiles, Physique & Instrumentation-Paris Observatory	Recipes description

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

The X-Shooter pipeline is a subsystem of the *VLT Data Flow System* (DFS). Its target user is ESO *Quality Control and Data Products Department* (QCDPD) in the generation of master calibration data, the reduction of scientific exposures, and data quality control. It should also serve as a quick look tool for *Paranal Science Operations* (PSO). Additionally, the X-Shooter pipeline recipes are made public to the user community to allow a more personalised processing of the data from the instrument. This document serves as a manual for X-Shooter data reduction with the X-Shooter pipeline.

This manual is a complete description of the data reduction recipes implemented by the the CPL based X-Shooter pipeline, reflecting the status of the X-Shooter pipeline as of Apr 29, 2013 (version 2.2.0).

1.2 Acknowledgements

The X-shooter pipeline 2.0 was developed at ESO by Andrea Modigliani and Daniel Bramich. It is based on the original Data Reduction Library (DRL) delivered with the instrument and written by P. Goldoni, F. Royer, R. Haigron, L. Guglielmi, P. Francois, M. Horrobin and H. Flores. The current release upgrades the slit mode data reduction.

This work would not have been possible without the contribution of a number of colleagues. Joel Vernet followed the DRL development and contributed to the review process. Paul Bristow contributed the physical model for the instrument calibration. Sabine Moehler worked on the flux calibration standards, response curve computation and OCA rules. Cesar Enrique Garcia Dabo and Wolfram Freudling contributed to the Reflex workflow.

We thank several beta testers and users who provided useful feedback. In particular, we would like to mention Ana Monreal Ibero, Lise Christensen, Frederik Schoenebeck, Pasquier Noterdaeme, Stephane Blondin and Guido Cupani.

1.3 Scope

This document describes the CPL based X-Shooter pipeline used at ESO-Garching and ESO-Paranal for the purpose of data assessment and data quality control. Updated versions of the present document may be found on [15]. For general information about the current instrument pipelines status we remind the user of [7]. Quality control information are at [6]. Additional information on CFITSIO, the Common Pipeline Library (CPL) and EsoRex can be found respectively at [18], [11], [13]. The Reflex and Gasgano front-ends are described in [10] and [14]. A description of the instrument is in [8], [9]. Additional information on the DFS and VLT data interfaces are in [4], [12], [5]. More information on the spectral format recovery procedure and on the physical model are in [1], [3], [2]. A short description of the pipeline is in [16].

1.4 Reference documents

[1] Reference objects identification: pattern-matching or first-guess model? A procedure to determine the spectral format of X-Shooter takes advantage of both techniques. https://websqa.hq.eso.org/sdd/pub/Pipeline/Publications/E31.ps. ADASS-XVII, 2008.

[2] Paul Bristow. *Application of the X-shooter physical model to flexure analysis*. ESO/INS, ADS. AN,332,2011, p224.

[3] Paul Bristow. *X-Shooter Physical model overview*. ESO/INS, ftp://ftp.eso.org/pub/dfs/pipelines/xsh/xsh_physmod_OV.pdf. VLT-MAN-ESO-14650-4840.

[4] ESO. *VLT Data Flow System Specifications for Pipeline and Quality Control*. VLT-SPE-ESO-19600-1233.

[5] ESO/DMO. *ESO DICB – Data Interface Control Document*. GEN-SPE-ESO-19400-0794 (3.0).

[6] ESO/DMO/DFO, *http://www.eso.org/observing/dfo/quality/*. *ESO-Data Flow Operation home page*.

[7] ESO/DMO/DFO, *http://www.eso.org/observing/dfo/quality/pipeline-status.html*. *X-Shooter Pipeline Current Status*.

[8] ESO/INS, *http://www.eso.org/sci/facilities/paranal/instruments/xshooter*. *X-shooter instrument home page*.

[9] ESO/INS, *http://www.eso.org/sci/facilities/paranal/instruments/xshooter/doc*. *X-shooter User Manual*.

[10] ESO/SDD/DFI, *http://www.eso.org/sci/software/reflex/*. *Reflex User Manual*. VLT-SPE-XXX-XXX.

[11] ESO/SDD/DFS, *http://www.eso.org/cpl/*. *CPL home page*.

[12] ESO/SDD/DFS. *DFS Pipeline & Quality Control – User Manual*. VLT-MAN-ESO-19500-1619.

[13] ESO/SDD/DFS, *http://www.eso.org/cpl/esorex.html*. *ESOREX home page*.

[14] ESO/SDD/DFS, *http://www.eso.org/gasgano/*. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932.

[15] ESO/SDD/DFS, *http://www.eso.org/pipelines*. *X-Shooter Pipeline Web Page*. VLT-MAN-ESO-14650-4840.

[16] A. Modigliani et al. *The X-shooter pipeline*. ESO/SDD, ADS. Proc. SPIE,7737-85,2010,p773728.

[17] A. Modigliani. *Detmon user manual*. ESO/SDD/PSD, ftp://www.eso.org/pipelines.

[18] NASA, *http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html*. *CFITSIO home page*.

2 Overview

In collaboration with instrument consortia, the Pipeline Systems Department (PSD) of the Software Development Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

Data quality control: pipelines are used to produce the quantitative information necessary to monitor instrument performance.

Master calibration product creation: pipelines are used to produce master calibration products (*e.g.*, combined bias frames, super-flats, wavelength dispersion solutions).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes (*e.g.*, combined ISAAC jitter stacks; bias-corrected, flat-fielded FORS images, wavelength-calibrated X-Shooter spectra). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. **In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.**

Instrument pipelines consist of a set of data processing modules that can be called from the command line, from the automatic data management tools available on Paranal, from Reflex, or from Gasgano.

ESO offers three front-end applications for launching pipeline recipes; namely *Reflex* [10], *Gasgano* [14] and *EsoRex*. The last two are included in the pipeline distribution (see Appendix C, page 138). These applications can also be downloaded separately from `www.eso.org/reflex`, `www.eso.org/gasgano`, `www.eso.org/cpl/esorex.html` respectively.

The X-Shooter instrument and the different types of X-Shooter raw frames and auxilliary data are described in Sections 4, 7, and 8.

A brief introduction to the usage of the available reduction recipes using Reflex, Gasgano or EsoRex is presented in Section 5. In Section 6 we advice the user about known data reduction problems.

An overview of the data reduction, the input data, and the recipes involved in the calibration cascade is provided in Section 9.

More details on inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in Section 10.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 11.

In Appendix C the installation of the X-Shooter pipeline recipes is described.

Throughout this manual we refer to the UV-Blue arm, the Visual-Red arm, and the Near-IR arm, of X-shooter as UVB, VIS and NIR, respectively.

3 What's new in pipeline release 2.2.0

This pipeline release marks the culmination of the project to improve the data products from the XSHOOTER pipeline for SLIT mode observations (i.e. excluding IFU). It includes the major enhancements that we list below. Please note that we also fixed bugs throughout the code, cleaned up the code and improved performance. Users should use this pipeline release from 2.0 on to reduce consistently their X-shooter data from scratch.

- Improved the noise model implementation.
- Implemented the correct propagation of errors during image combination and at the rectification stage.
- Replaced all code associated with bad pixel propagation. Pixels are now flagged with bit-codes at various stages in the pipeline and the results are stored in the third extension of the FITS files (we have dropped the separate bad pixel map files used in previous versions). Pixel flags are now propagated using a bitwise-OR in the correct fashion. The pixel flags may be decoded into good and bad pixels via the use of the general recipe parameter **decode-bp** allowing the user the flexibility to choose how to treat bad pixels. We implemented the correct flagging of saturated, negative or extrapolated pixel values in the raw images.
- Improved the overscan bias level correction.
- Redesigned xsh_mbias and replaced the algorithm for combining bias frames.
- Redesigned xsh_mdark and replaced the algorithm for combining dark frames.
- Redesigned xsh_mflat and replaced the algorithm for combining flat frames.
- Replaced the algorithm for inter-order background determination.
- Optimised the parameters for arc line detection.
- Created a more complete and uniform reference arc line list.
- Produced new physical model configuration files.
- Reduced the systematic errors present in the wavelength solution in UVB and VIS.
- Improved response determination.
- Dropped many unused parameters in the response/science recipes.
- Modified the order of operations in the response/science recipes.
- Improved the code for cosmic ray detection via Laplacian edge detection.
- Replaced the algorithm for standard extraction allowing for interpolation over bad pixels using a locally determined spatial profile.
- The manual has been rewritten with a focus on describing properly the recipe inputs/outputs/parameters and the algorithms that are employed.

This pipeline release implements –with respect to release 2.0.0– the following improvements:

- Improved flux calibration with provision of a new set of more accurate spectral std models, implementation of a telluric correction during response computation, optimization of the definition of regions where to mask (in VIS,NIR) tellurics, changed default response interpolation of raw response from a cubic spline smooth to a cubic spline fit of a set of data points that can be modified by the user, fixed a problem in the computation of response and flux calibration for NIR-JH band data.
- Improved OCA rules for reflex based data reduction.
- Report re-sampled spectra accuracy on science merged spectra.
- Additional improvements on the reflex interactive windows. layout.

4 X-Shooter Instrument Description

X-shooter is a single target spectrograph for the Cassegrain focus of one of the VLT UTs (See Figure 4.1). The instrument covers in a single exposure the spectral range from the UV to the K' band. It is designed to maximize the sensitivity in this spectral range through the splitting of light into three arms with optimized optics, coatings, dispersive elements and detectors. It operates at intermediate resolutions (R=3000-17000, depending on wavelength and slit width) with fixed echelle spectral format (with prism cross-dispersers) in the three arms. The layout and the small number of moving functions (and therefore instrument modes) make the instrument simple and easy to operate and permit a fast response. The possibility to observe in a single shot faint sources at the sky limit with an unknown flux distribution has inspired the name of the instrument. More details are in [8].

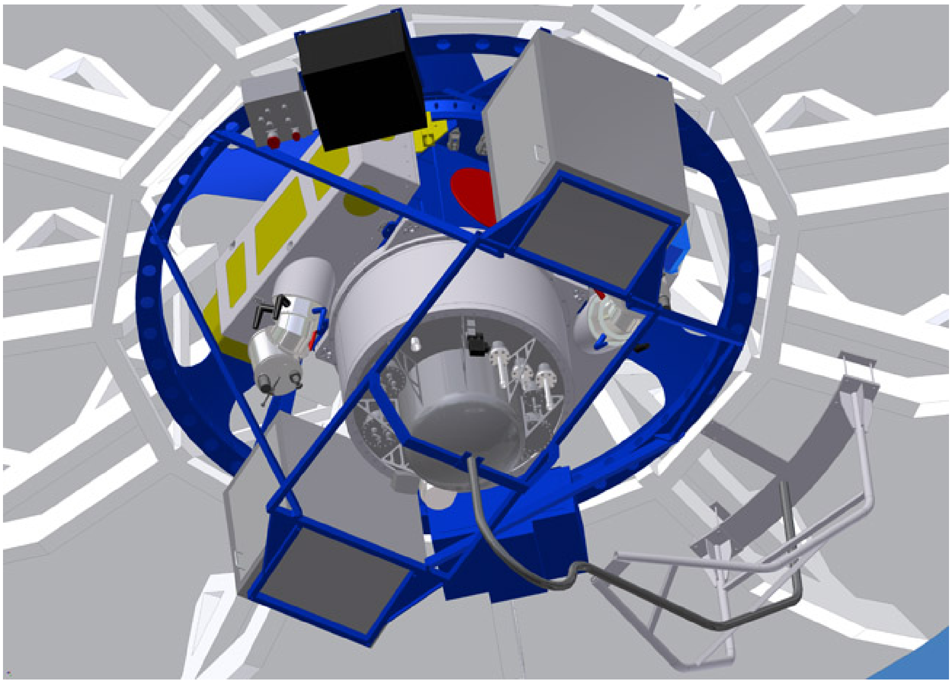


Figure 4.1: *A schematic view of X-shooter and its subsystems.*

4.1 Instrument Description

The instrument concept is illustrated in the schematic view below (Figure 4.3).

The calibration lamps are located in the upper section of the instrument. Mechanical slides can insert above or at the focal plane calibration lamp mirrors, a small integral field unit reformatting a 1.8" x 4" area into a slit of 0.6" x 12" or mirrors feeding an acquisition and guiding camera. After the telescope focal plane, the light beam is split into three spectral ranges (the UV-Blue arm - referred to as UVB, the Visual-Red arm - referred to as VIS, and the Near-IR arm - referred to as NIR) by two dichroics and focused by auxiliary optics on three separate slits.

Wavelength range	300-2500 nm split over 3 arms
UV-Blue arm	Range: 300-550 nm in 11 orders
Resolution:	4500 (1" slit)
Detector:	4k x 2k E2V CCD
Visual-red arm	Range: 550-1000 nm in 14 orders
Resolution:	7000 (1" slit)
Detector:	4k x 2k MIT/LL CCD
Near-IR arm	Range: 1000-2500 nm in 16 orders
Resolution:	4500 (1" slit)
Detector:	2k x 1k Hawaii 2RG
Slit length	12"
Beam separation	Two high efficiency dichroics
Atmospheric dispersion compensation	In the UV-Blue and Visual-red arms
Integral field unit	1.8" x 4" reformatted into 0.6" x 12"

The maximum slit length is 12 arcsec for all arms. The transfer optics in the UV-Blue and Visual-Red arms include atmospheric dispersion correctors. All three arms include piezo mirrors for flexure compensation. The three spectrograph arms each include fixed echelle grating and prisms cross-dispersers, providing full spectral coverage in a single exposure with a spectral resolution between 3000 and 17000 depending on the slit width and the spectral arm. In Figure 4.2 we show ThAr calibration frames for each arm.

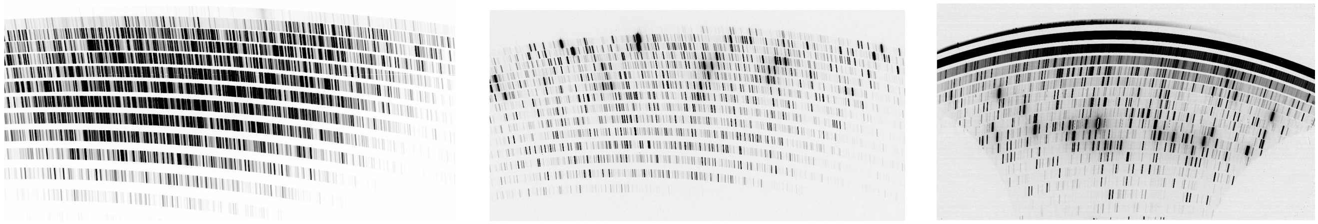


Figure 4.2: ThAr calibration frame for each arm of X-shooter (UVB on the left, VIS in the center, NIR on the right).

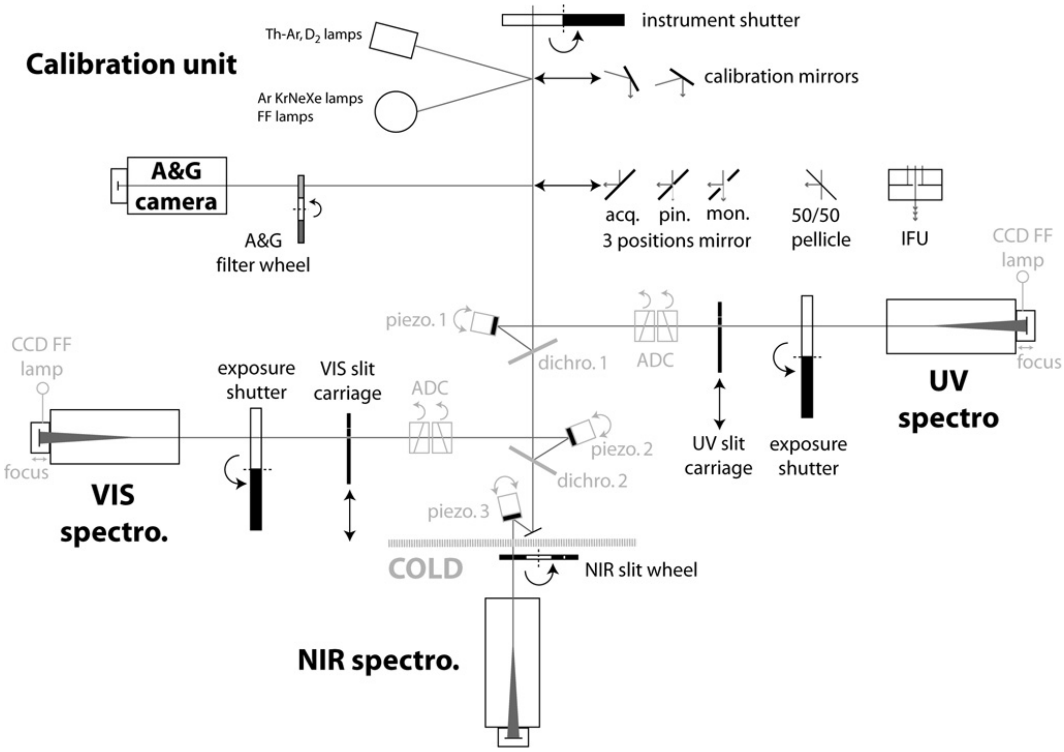


Figure 4.3: *A schematic view of X-shooter and its subsystems.*

5 Quick start

This section describes how to make immediate usage of the X-Shooter pipeline recipes. Please note the convention in this manual for referring to the three X-shooter arms as simply UVB, VIS and NIR.

5.1 X-Shooter pipeline recipes

The current X-Shooter pipeline is based on a set of 20 stand-alone recipes involved in the data reduction cascade, and two other plugins.

```
xsh_lingain           : Computes a detector's gain/linearity-map
xsh_mbias             : Creates a master bias frame
xsh_mdark             : Creates a master dark frame
xsh_predict           : Computes a first guess dispersion solution and order trace
xsh_orderpos          : Creates the orders centre traces table file
xsh_mflat             : Creates the master flat and the order edge traces table frames
xsh_2dmap             : Creates a wavelength and spatial resampling solution, including
                          a clean arc line list
xsh_flexcomp          : Computes instrument flexures
xsh_wavecal           : Computes arc lines tilt and instrument resolution
xsh_respon_slit_stare : Computes the response function in SLIT stare mode
xsh_respon_slit_offset: Computes the response function in SLIT and offset mode
xsh_respon_slit_nod   : Computes the response function in SLIT and nod mode
xsh_scired_slit_stare : Reduces a science exposure in SLIT configuration and stare mode
xsh_scired_slit_offset: Reduces a science exposure in SLIT configuration and offset mode
xsh_scired_slit_nod   : Reduces a science exposure in SLIT configuration and NOD mode
xsh_scired_ifu_stare  : Reduces a science IFU stare exposure and builds a 3D cube
xsh_scired_ifu_offset : Reduces a science IFU on-off exposure and builds a 3D cube

xsh_geom_ifu : Produces the spatial geometry of the IFU pattern on the Sky
xsh_scired_ifu_stare_drl  : Reduce science exposure in IFU configuration and stare mode
                            with atmospheric dispersion correction
xsh_scired_ifu_offset_drl  : Reduce science exposure in IFU configuration and offset mode
                             with atmospheric dispersion correction

xsh_util_physmod      : Generates physical model products
xsh_cfg_recover       : Optimizes a model configuration to match
                          data taken after a major format change
```

5.2 An introduction to Reflex, Gasgano and EsoRex

Before being able to call pipeline recipes to process a set of data, the data must be correctly classified, and associated with the appropriate calibrations. The *Data Classification* consists of tasks such as: "What kind of data am I?", *e.g.*, BIAS, "To which group do I belong?", *e.g.*, to a particular Observation Block or template. *Data Association* is the process of selecting appropriate calibration data for the reduction of a set of raw science frames. Typically, a set of frames can be associated if they share a number of properties, such as instrument and detector configuration. Since all the required information is stored in the FITS headers, data association is based on a set of header keywords (called "association keywords") and the process is specific to each type of calibration. The process of data classification and association is known as data organisation.

An instrument pipeline consists of a set of data processing modules that can be called from different host applications, namely:

- *Reflex* is a graphical tool that helps the user to execute data reduction workflows which contain several recipes. This dramatically decreases the time the user needs to run a whole reduction chain, from calibration and raw data down to the final products. *Reflex* takes care of grouping the different data sets, associating the calibration frames and managing the interdependencies between recipes in the calibration cascade. **Reflex is the recommended software tool for reducing your data.**
- *Gasgano* is an alternative data management tool that simplifies the data organization process. In addition, *Gasgano* allows the user to execute directly the pipeline recipes on a set of selected files.
- *EsoRex* is a command line tool used to run the pipeline recipes. *EsoRex* commands can be easily scripted.
- The Paranal observatory implements automatic data management tools that trigger the execution of pipeline recipes. This aspect is not covered in this manual.

5.3 Example of data reduction using the Reflex-based X-shooter workflow

For the user who is keen on starting reductions without being distracted by detailed documentation, we describe the steps to be performed to reduce the science data provided in the X-shooter demo data set supplied with the `Reflex 2.3` release. By following these steps, the user should have enough information to attempt a reduction of his/her own data without any further reading:

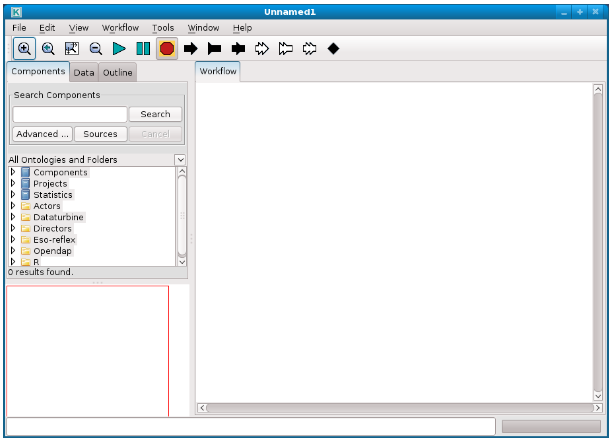


Figure 5.1: *The empty* `Reflex` *canvas.*

1. Start the `Reflex` application:

   ```
   reflex &
   ```

   The empty `Reflex` canvas as shown in Figure 5.1 will appear.

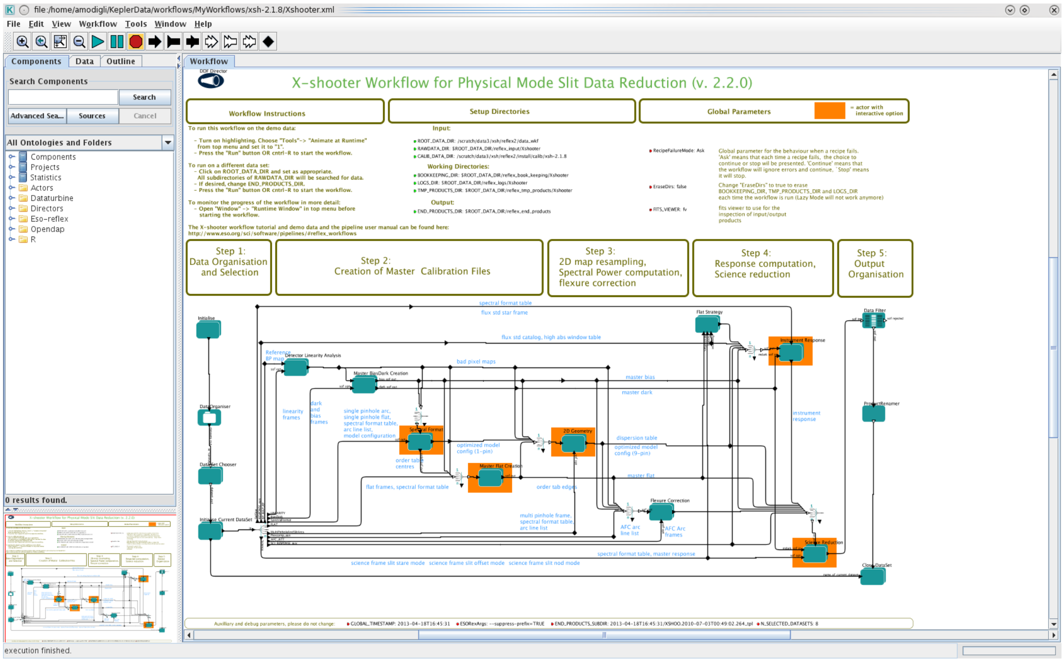


Figure 5.2: *X-shooter workflow general layout.*

2. Now open the X-shooter workflow by clicking on `File -> Open File`, selecting the file `xsh-2.2.0/Xshooter.xml` in the file browser. You will be presented with the workflow canvas shown in Figure 5.2. Note that the workflow will appear as a canvas in a new window.

3. To aid in the visual tracking of the reduction cascade, it is advisable to use component (or actor) highlighting. Click on `Tools -> Animate at Runtime`, enter the number of milliseconds representing the animation interval (1 ms is recommended), and click `OK`.

4. Under “Setup Directories” in the workflow canvas there are seven parameters that specify important directories (green dots). Setting the value of `ROOT_DATA_DIR` is the only necessary modification if you want to process data other than the demo data[1], since the value of this parameter specifies the working directory within which the other directories are organised. Double-click on the parameter `ROOT_DATA_DIR` and a pop-up window will appear allowing you to modify the directory string, which you may either edit directly, or use the `Browse` button to select the directory from a file browser. When you have finished, click `OK` to save your changes.

5. Click the button to start the workflow

6. The workflow will highlight the `Data Organiser` actor which recursively scans the raw data directory (specified by the parameter `RAWDATA_DIR` under “Setup Directories” in the workflow canvas) and constructs the DataSets. Note that the calibration and reference data must be present either in `RAWDATA_DIR` or in `CALIB_DATA_DIR`, otherwise DataSets may be incomplete and cannot be processed.

[1] If you used the install script `install_reflex`, then the value of the parameter `ROOT_DATA_DIR` will already be set correctly to the directory where the demo data was downloaded.

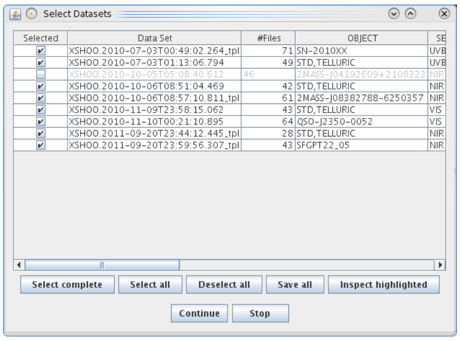


Figure 5.3: *The "Select Datasets" pop-up window. On purpose the demo includes an incomplete (grey) DataSet.*

7. The `Data Set Chooser` actor will be highlighted next and will display a "Select Datasets" window (see Figure 5.3) that lists the DataSets along with the values of a selection of useful header keywords, namely the object name, X-shooter arm, observing mode (SLOT/IFU), observing technique (STARE/NODDING/OFF detector read mode (for UVB/VIS detectors only), slit widths (`INS.OPTI.<num>.NAME` for `num=3/4/5` corresponding to UVB/VIS/NIR, respectively), exposure time, DIT (defined for NIR data only), and OB number. The first column consists of a set of tick boxes which allow the user to select the DataSets to be processed, and by default all DataSets are selected.

   You will find that one DataSet is grey and has no tick box. This is an incomplete DataSet provided to demonstrate how such DataSets look in the "Select Datasets" window. Moving the mouse over this grey DataSet will give you the information which kind of files are missing.

8. Click the `Continue` button and watch the progress of the workflow by following the red highlighting of the actors. A window will show which DataSet is currently being processed.

9. When the workflow has finished executing the pipeline recipe `xsh_respon_slit_offset` in the `Instrument Response` actor for the first DataSet, an interactive window will appear which shows a plot of the extracted and merged standard star spectrum in the top panel (and the response curve, if the standard star is found in the catalogue). Using the buttons at the top of this window, one may pan and zoom in on the spectrum in order to inspect absorption/emissions lines and other interesting spectral features. Below the extracted spectra the 2-dimensional rectified data are displayed - sky-corrected and sky (the latter only for STARE and OFFSET data).

10. When the workflow has finished executing the final pipeline recipe `xsh_scired_slit_stare/nod/offset` in the `Spectrum Reduction` actor for the first DataSet, an interactive window will appear (see Figure 5.4) which shows a plot of the extracted and merged spectrum in the top panel (and the flux-calibrated version, if a suitable instrument response curve exists, in the panel below. In the canvas appears no flux calibrated spectrum, as no suitable standard exist). Below the extracted spectra the 2-dimensional rectified data are displayed - sky-corrected and sky (the latter only for STARE and OFFSET data).

11. Click on the `Continue wkf` button and the workflow will write out the important products of the reduction cascade to the end products directory (specified by the parameter `END_PRODUCTS_DIR` under "Setup Directories" in the workflow canvas), which includes the extracted and merged spectrum.

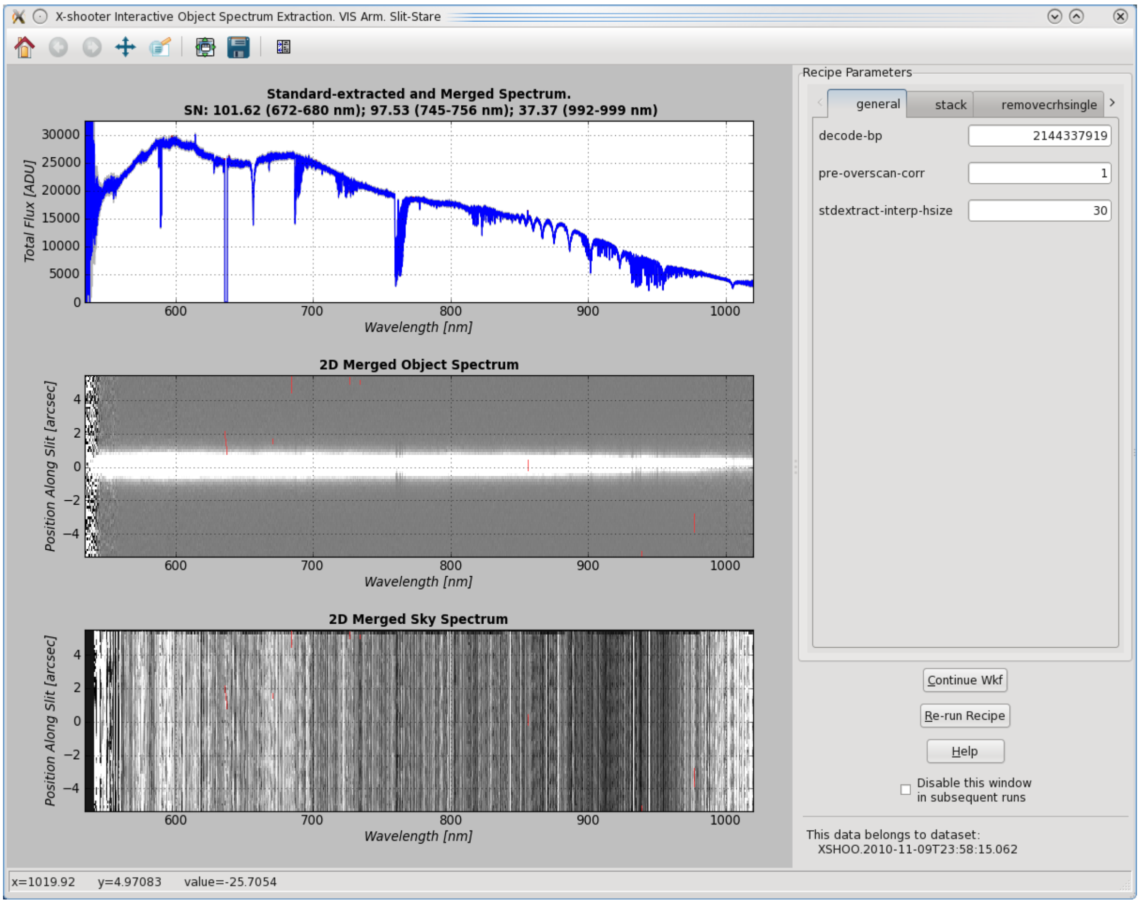


Figure 5.4: *The interactive pop-up window for the* `Science Reduction` *actor and X-shooter pipeline recipe* `xsh_scired_slit_stare`*. The extracted and merged spectrum for the VIS DataSet is displayed in the top panel.*

12. The workflow will automatically move on to the next DataSet repeating the reduction cascade and displaying the interactive windows with the response curves and extracted and merged spectra, respectively. For each DataSet the procedure is the same; simply inspect the response curve and the science spectrum and then continue until all data sets are processed.

5.3.1 Using Gasgano

Another convenient tool useful for familiarizing oneself with the X-Shooter pipeline recipes and their usage is the graphical user interface *Gasgano*. It provides a complete graphical user interface for data browsing, classification and association, and offers several other utilities such as easy access to pipeline recipes, documentation and the preferred data display tools.

Gasgano can be started from the system prompt in the following way:

```
gasgano $HOME/gasgano/config/XSH.prefs &
```

where we have explicitly passed as first optional argument the X-Shooter preferences file which defines proper defaults for X-Shooter data reduction. The user may prefer to set a UNIX alias accordingly. The *Gasgano* main window will appear.

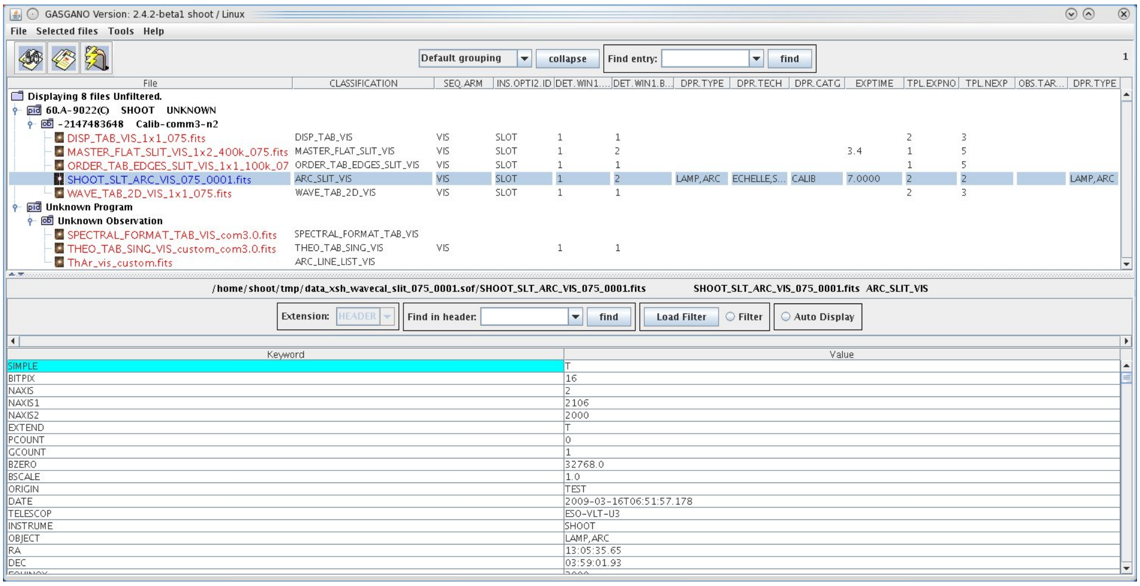


Figure 5.5: *The Gasgano main window.*

In Figure 5.5, a view of a set of X-Shooter data is shown as an example. *Gasgano* can be configured to point to the directories where the data to be handled are located using the navigation panels accessible via the *Add/Remove Files* entry of the *File* menu (shown on the upper left of the figure).

The data are hierarchically organised as preferred by the user. Next to each file name, the classification field and the values of the following FITS keywords are shown (we omit the prefix HIERARCH.ESO):

Keyword name	Purpose
CLASSIFICATION	Data classification
DPR.TYPE	Data processing type
DATE	Observing date
SEQ.ARM	Selected arm
DET.READ.CLOCK	Detector read out clock (speed and binning setting in UVB/VIS)
EXPTIME	Exposure time
DET.DIT	Detector DIT (NIR)
DET.NDIT	Detector repetitions of DIT (NIR)
INS.OPTI2.ID	SLIT/IFU
INS.OPTI3.NAME	SLIT setup (width$\times$ length) for UVB
INS.OPTI4.NAME	SLIT setup (width$\times$ length) for VIS
INS.OPTI5.NAME	SLIT setup (width$\times$ length) for NIR
OBS.TARG.NAME	Observation Block target name

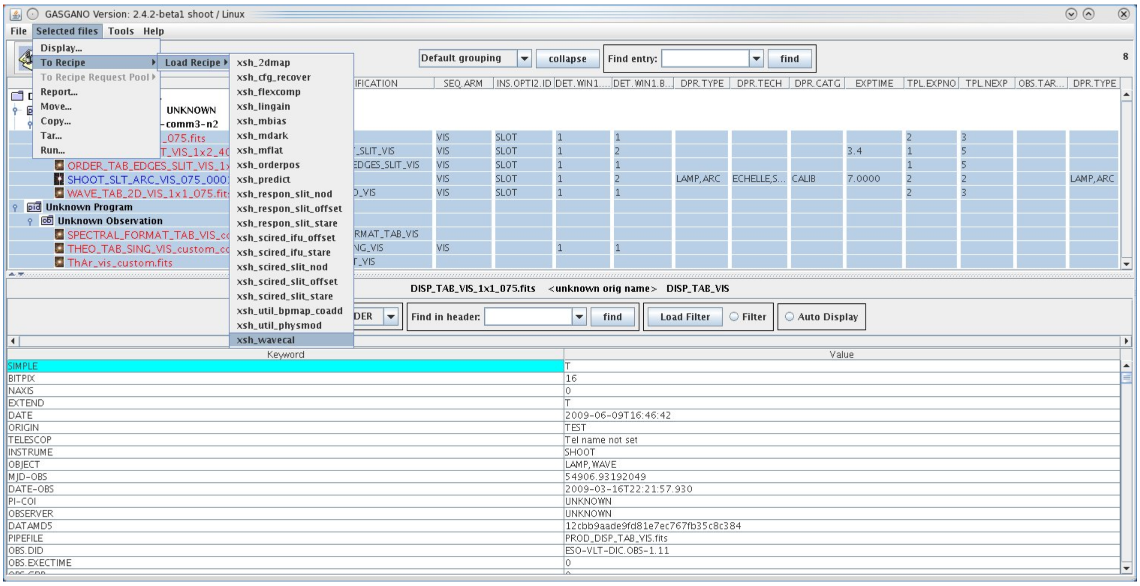


Figure 5.6: *Selecting files to be processed by the X-Shooter pipeline xsh_2dmap recipe*

The CLASSIFICATION field provides either the value of the PRO.CATG header keyword, for pipeline products; or a user defined file classification, if provided, defined in the classification rule file (which can be accessed by Gasgano from the Tools –> Classification rules... tab); or the default value "UNDEFINED". File classification rules are selection rules which assign to a FITS file a classification based on the value of a few FITS header keywords, usually the DPR.TYPE, DPR.TECH, DPR.CATG values, which respectively define the file data type, acquisition technique and category. Additional relevant keywords are OBS.TARG.NAME, EXPTIME, DATE, SEQ.ARM, INS.OPTI2.ID, INS.OPTIi.NAME (i=3-5), DET.READ.CLOCK, DET.DIT, DET.NDIT.

Additional useful keywords are:

Keyword name	Purpose
ORIGFILE	Frame original file (describes frame type)
INS.LAMP1.ST	Lamp status (ThAr - Spectral UVB/VIS)
INS.LAMP2.ST	Lamp status (ArKeNeXe - penray - Spectral NIR)
INS.LAMP3.ST	Lamp status (UVB_LOW_D2 - Flat Field UVB low)
INS.LAMP4.ST	Lamp status (UVB_High - Flat Field UVB/High)
INS.LAMP5.ST	Lamp status (FF VIS - Flat Field VIS)
INS.LAMP6.ST	Lamp status (FF NIR - Flat Field NIR)

More information about a single frame can be obtained by left-clicking on its name: the corresponding FITS file header will be displayed on the bottom panel, where specific keywords can be filtered and searched. Images and tables may be easily displayed using the viewers specified in the appropriate *Preferences* fields. Such a field allows also to set the file filter, which should point to the $HOME/gasgano/config/xsh.oca file. This rule file provides classification rules to select X-Shooter data corresponding to a given standard data reduction setting.

Frames can be selected from the main window for being processed by the appropriate recipe: in Figure 5.6, a multi-pinhole frame, previously produced master bias and master flat frames, together with a line guess table, an order table, and a theoretical map table are all selected and sent to the *xsh_2dmap* recipe. This will open a *Gasgano* recipe execution window (see Figure 5.7), having all the specified files listed in its *Input Frames* panel.

Help about the recipe is available from the *Help* menu. Before launching the recipe, its configuration may be modified on the *Parameters* panel (on top). The window contents might be saved for later use by selecting the *Save Current Settings* entry from the *File* menu, as shown in Figure 5.7.

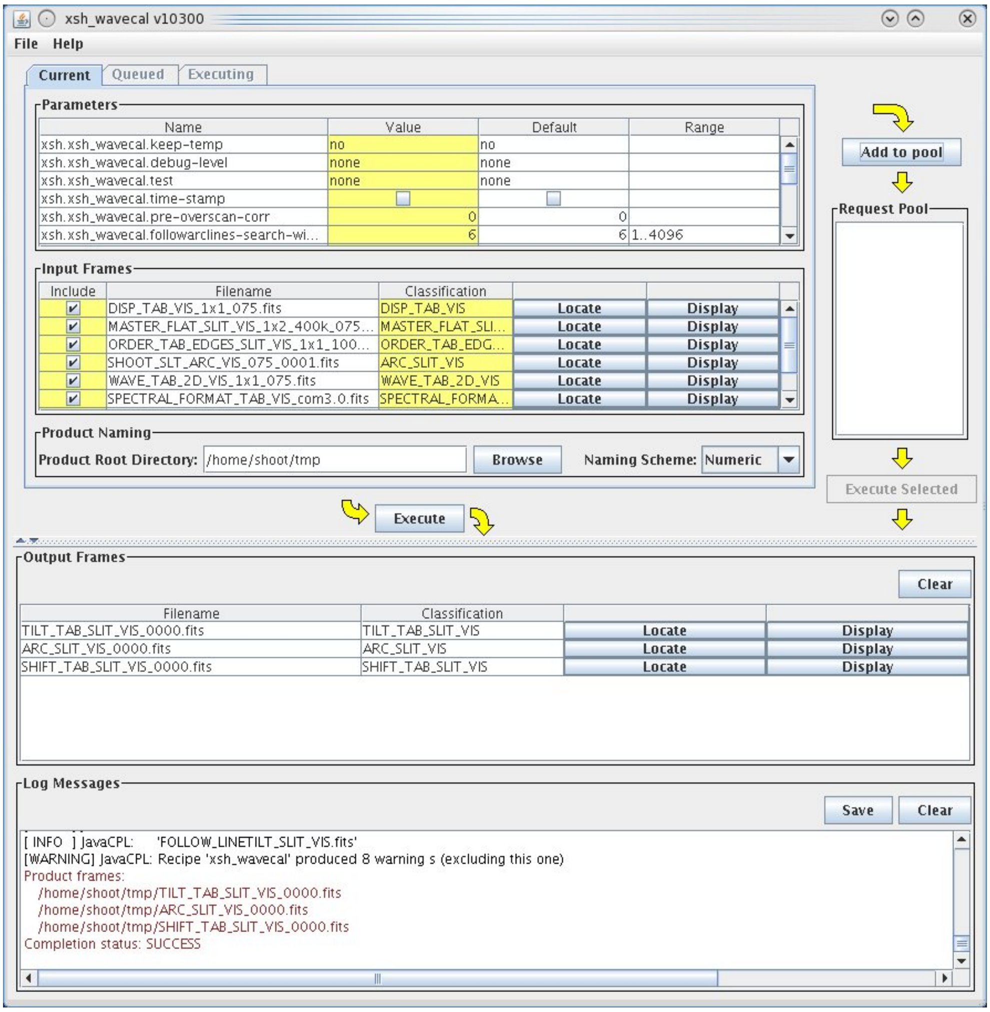


Figure 5.7: *The Gasgano recipe execution window.*

At this point the recipe can be launched by pressing the *Execute* button. Messages from the running recipe will appear on the *Log Messages* panel at the bottom, and in case of successful completion, the products will be listed in the *Output Frames* panel, where they can be easily viewed and located back in the Gasgano main window. Please refer to the *Gasgano User's Manual* [14] for a more complete description of the *Gasgano* interface.

5.3.2 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. A disadvantage of EsoRex is that it does not offer all the facilities available with *Reflex*, and the user must classify and associate the data using the information contained in the FITS header keywords (see Section 7, page 37). The user is also responsible for defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

The set-of-frames: Each pipeline recipe is run on a set of input FITS data files. When using *EsoRex* the file names must be listed together with their DO category [2] in an ASCII file called the *set-of-frames* (SOF), that is required when launching a recipe. [3] DO categories for the supported X-Shooter input raw frames are indicated in section 7.12.

Here is an example SOF, valid for the *xsh_wavecal* recipe (in polynomial mode):

```
/path_raw/SHOOT_SLT_ARC_VIS_063_0004.fits     ARC_SLIT_VIS
/path_ref/SPECTRAL_FORMAT_TAB_VIS.fits        SPECTRAL_FORMAT_TAB_VIS
/path_ref/ARC_LINE_LIST_VIS.fits              ARC_LINE_LIST_VIS
/path_cdb/MASTER_BIAS_VIS.fits                MASTER_BIAS_VIS (optional)
/path_cdb/ORDER_TAB_EDGES_SLIT_VIS.fits       ORDER_TAB_EDGES_SLIT_VIS
/path_cdb/WAVE_TAB_2D_VIS.fits                WAVE_TAB_2D_VIS
/path_cdb/DISP_TAB_2D_VIS.fits                DISP_TAB_2D_VIS
/path_cdb/THEO_TAB_SING_VIS.fits              THEO_TAB_SING_VIS
```

It contains for each input frame the full path file name and its DO category. Here `/path_raw//path_ref/` `/path_cdb//path_pro/` indicate the full path to directories with raw, reference, master calibrations and other pipeline products. The pipeline recipe will access the listed files when required by the reduction algorithm.

Note that the X-Shooter pipeline recipes do not verify the correctness of the classification tags specified by the user in the SOF. In the above example, the recipe *xsh_wavecal* will treat the frame `/path_raw/SHOOT_SLT_ARC_VIS_063_0004.fits` as an `ARC_SLIT_VIS`, the frame `/path_cdb/ORDER_TAB_EDGES_SLIT_VIS.fits` as an `ORDER_TAB_EDGES_SLIT_VIS`, etc., even when they do not contain this type of data. The recipe will also assume that all frames are associated correctly, *i.e.*, that they all come from the same arm, dichroic and bin setting.

The reason of this lack of control is that the X-Shooter recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Reflex* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 5.3). Also this lack of control allows the user to reduce e.g. an arc lamp frame pretending it is a science frame.

A recipe handling an incorrect SOF may stop or display unclear error messages at best. In the worst cases, the recipe would apparently run without any problem, producing results that may look reasonable, but are actually flawed.

EsoRex syntax: The basic syntax for using EsoRex is the following:

esorex [esorex_options] recipe_name [recipe_options] set_of_frames

[2] The indicated *DO category* is a label assigned to any data type after it has been classified, which is then used to identify the frames listed in the *set-of-frames*

[3] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 5.7, page 26).

To get more information on how to customise EsoRex (see also [13]) run the command:

esorex - -help

To generate a configuration file esorex.rc in the directory $HOME/.esorex run the command:

esorex - -create-config

A list of all available recipes, each with a one-line description, can be obtained using the command:

esorex - -recipes

EsoRex searches for recipes in the directory specified by the option

esorex - -recipe-dir=installation_directory

All recipe parameters (aliases) and their default values can be displayed by the command

esorex - -params recipe_name

To obtain a brief description of each parameter for a specific pipeline recipe, execute the command:

esorex - -help recipe_name

To obtain more details about a given recipe, issue the following command:

esorex - -man-page recipe_name

Recipe configuration: To each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe.[4] The configuration files are normally generated in the directory `$HOME/.esorex`, and have the same name as the recipe to which they are related, with the filename extension `.rc`. For instance, the recipe *xsh_wavecal* has its *EsoRex* generated configuration file named `xsh_wavecal.rc`, and is generated with the command:

esorex - -create-config xsh_wavecal

The definition of one parameter of a recipe may look like this:

```
# --wavemap-deg-y
# Y degree of polynomial wavelength map.
xsh.xsh_wavecal.wavemap-deg-y=3
```

In this example, the parameter `xsh.xsh_wavecal.wavemap-deg-y` is set to the value `3`. In the configuration file generated by *EsoRex*, one or more comment lines are added containing information about the possible values of the parameter, and an alias that could be used as a command line option.

A recipe configuration file different from the default one may be specified on the command line:

esorex - -recipe-config=my_alternative_recipe_config

Recipe parameters are defined in Section 10 and their role is described in Section 11.

More than one configuration file may be maintained for the same recipe but, in order to be used, a configuration file not located under `$HOME/.esorex`, or having a name different from the recipe name, should be explicitly specified when launching a recipe.

[4]The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 5.7, page 26).

Recipe execution: A recipe can be run by giving its name to *EsoRex*, together with the name of a set-of-frames. For instance, the following command line would be used to run the recipe *xsh_wavecal* for processing the files specified in the set-of-frames `xsh_wavecal.sof`:

esorex xsh_wavecal xsh_wavecal.sof

The recipe parameters may be modified either by editing directly the configuration file that is used, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name and before the SOF name, and they will supersede the system defaults and/or the configuration file settings. For instance, to set the *xsh_wavecal* recipe *wavemap-deg-y* parameter to `4`, the following should be entered:

esorex xsh_wavecal - -wavemap-deg-y=4 xsh_wavecal.sof

Every recipe provides a parameter *debug-level* which – when enabled – causes intermediate results to be saved to the local directory. This allows more detailed inspection of the recipe execution:

esorex xsh_wavecal - -debug-level=medium xsh_wavecal.sof

For more information on *EsoRex*, see `www.eso.org/cpl/esorex.html`.

5.4 Reduction chains using EsoRex

In the following sections, we elaborate the default reduction chains using the pipeline recipes with default parameters. If the user sets the parameter **keep-temp** to 'yes', then other products are generated, either internal products or additional ones used for debugging purposes. To keep the documentation at a reasonable size, we have decided in this release not to describe them.

There are two main data reduction modes:

- physical model, where solutions are obtained by optimizing the instrument physical model parameters contained in a table (XSH_MOD_CFG_TAB_ARM) to the data.
- poly, where solutions are obtained via polynomial fits to the data, starting from reference tables that can be generated with the recipe xsh_util_physmod, starting from a model configuration file corresponding to the current data set and a reference line table.

We recommend using the data reduction chain corresponding to the physical model which is the reduction mode supported by Reflex. Furthermore poly mode is not supported by Operations at ESO and therefore it is not well tested.

To obtain proper quality the user has to repeat the full data reduction chain on a coherent data set using proper reference and calibration frames (proper instrument setting, closest in time to the observation) and, to reduce data taken after 1st October 2009, use the model configuration files named as xsh_arm_def_aug10.fits (arm=uvb, vis, nir) provided in the kit release and deployed during installation in the directory <install_dir>/calib/xsh-2.2.0/ifu/cal/ , where install_dir is the user chosen directory during kit installation.

5.4.1 Reduction chain for "physical" mode for calib slit data

esorex xsh_lingain xsh_lingain.sof
esorex xsh_mbias xsh_mbias.sof
esorex xsh_mdark xsh_mdark.sof
esorex xsh_predict xsh_predict.sof
esorex xsh_orderpos xsh_orderpos.sof
esorex xsh_mflat xsh_mflat.sof
esorex xsh_2dmap xsh_2dmap.sof
esorex xsh_flexcomp xsh_flexcomp.sof
esorex xsh_wavecal xsh_wavecal.sof
esorex xsh_respon_slit_stare xsh_respon_slit_stare.sof
esorex xsh_respon_slit_offset xsh_respon_slit_offset.sof
esorex xsh_scired_slit_stare xsh_scired_slit_stare.sof
esorex xsh_scired_slit_offset xsh_scired_slit_offset.sof
esorex xsh_scired_slit_nod xsh_scired_slit_nod.sof

5.4.2 Reduction chain for "poly" mode for calib slit data

esorex xsh_lingain xsh_lingain.sof
esorex xsh_util_physmod xsh_util_physmod.sof
esorex xsh_mbias xsh_mbias.sof
esorex xsh_mdark xsh_mdark.sof
esorex xsh_predict xsh_predict.sof
esorex xsh_orderpos xsh_orderpos.sof
esorex xsh_mflat xsh_mflat.sof
esorex xsh_2dmap xsh_2dmap.sof
esorex xsh_flexcomp xsh_flexcomp.sof
esorex xsh_wavecal xsh_wavecal.sof
esorex xsh_respon_slit_stare xsh_respon_slit_stare.sof
esorex xsh_respon_slit_offset xsh_respon_slit_offset.sof
esorex xsh_scired_slit_stare xsh_scired_slit_stare.sof
esorex xsh_scired_slit_offset xsh_scired_slit_offset.sof
esorex xsh_scired_slit_nod xsh_scired_slit_nod.sof

5.4.3 Reduction chain for "physical" mode for calib IFU data (operational)

esorex xsh_lingain xsh_lingain.sof
esorex xsh_mbias xsh_mbias.sof
esorex xsh_mdark xsh_mdark.sof
esorex xsh_predict xsh_predict.sof
esorex xsh_orderpos xsh_orderpos.sof
esorex xsh_mflat xsh_mflat.sof
esorex xsh_2dmap xsh_2dmap.sof

esorex xsh_flexcomp xsh_flexcomp.sof
esorex xsh_wavecal xsh_wavecal.sof
esorex xsh_scired_ifu_stare xsh_scired_ifu_stare.sof
esorex xsh_scired_ifu_offset xsh_scired_ifu_offset.sof

5.4.4 Reduction chain for "poly" mode for calib IFU data (operational)

esorex xsh_lingain xsh_lingain.sof
esorex xsh_util_physmod xsh_util_physmod.sof
esorex xsh_mbias xsh_mbias.sof
esorex xsh_mdark xsh_mdark.sof
esorex xsh_predict xsh_predict.sof
esorex xsh_orderpos xsh_orderpos.sof
esorex xsh_mflat xsh_mflat.sof
esorex xsh_2dmap xsh_2dmap.sof
esorex xsh_flexcomp xsh_flexcomp.sof
esorex xsh_wavecal xsh_wavecal.sof
esorex xsh_scired_ifu_stare xsh_scired_ifu_stare.sof
esorex xsh_scired_ifu_offset xsh_scired_ifu_offset.sof

5.4.5 Reduction chain for "physical" mode for calib IFU data (with atmospheric dispersion correction)

esorex xsh_lingain xsh_lingain.sof
esorex xsh_mbias xsh_mbias.sof
esorex xsh_mdark xsh_mdark.sof
esorex xsh_predict xsh_predict.sof
esorex xsh_orderpos xsh_orderpos.sof
esorex xsh_mflat xsh_mflat.sof
esorex xsh_2dmap xsh_2dmap.sof
esorex xsh_flexcomp xsh_flexcomp.sof
esorex xsh_wavecal xsh_wavecal.sof
esorex xsh_geom_ifu xsh_geom_ifu.sof
esorex xsh_scired_ifu_stare_drl xsh_scired_ifu_stare_drl.sof
esorex xsh_scired_ifu_offset_drl xsh_scired_ifu_offset_drl.sof

5.4.6 Reduction chain for "poly" mode for calib IFU data (with atmospheric dispersion correction)

esorex xsh_lingain xsh_lingain.sof
esorex xsh_util_physmod xsh_util_physmod.sof
esorex xsh_mbias xsh_mbias.sof
esorex xsh_mdark xsh_mdark.sof
esorex xsh_predict xsh_predict.sof
esorex xsh_orderpos xsh_orderpos.sof

esorex xsh_mflat xsh_mflat.sof
esorex xsh_2dmap xsh_2dmap.sof
esorex xsh_flexcomp xsh_flexcomp.sof
esorex xsh_wavecal xsh_wavecal.sof
esorex xsh_geom_ifu xsh_geom_ifu.sof
esorex xsh_scired_ifu_stare_drl xsh_scired_ifu_stare_drl.sof
esorex xsh_scired_ifu_offset_drl xsh_scired_ifu_offset_drl.sof

6 Known problems

The following is a list of currently-known problems with the X-Shooter pipeline recipes, and workarounds, if available. In the list we indicate the recipe, the problem, and the proposed solution. `Reflex` specific problems are indicated in the corresponding tutorial. In case the user finds further problems, we kindly ask to contact usd-help@eso.org with a full recipe log, the exact EsoRex command given (including parameter values), the set of frames and all the input data (via ftp or on a given accessible URL).

We suggest that the user employs `Reflex` for the data reduction.

This release includes supports for IFU data reduction (operational, without atmospheric dispersion correction). Tests on standard star observations in stare mode show that we currently reach a relative slitlet alignment accuracy of the order of half a detector pixel. This release includes also support for an alternative (interactive) EsoRex based IFU data reduction, via recipes xsh_geom_ifu, xsh_scired_ifu_stare_drl, xsh_scired_ifu_offset_drl, that allows to correct for effects of atmosphere dispersion and reach higher accuracy.

The last order of the NIR arm -longer wavelengths of the K -band- is partly vignetted in the instrument. The vignetting increases in the direction orthogonal to the dispersion and reaches a maximum value of 10% closer to the border of the detector. The effect can be verified tracing sky emission lines or the flat field.

- xsh_lingain:
  - Possible problem: There are no linearity data in period P84 (01/10/2009 - 31/03/2010).
  - Proposed solution: The user may use either more recent data or the reference bad pixel maps distributed with the kit release.
  - We found that this recipes does not properly detect non-linear pixels. The user is recommended not to use the corresponding bad pixel map in the data reduction chain. The non-linear pixels are detected as part of the master flat creation.
- xsh_2dmap:
  - Possible problem:
    The recipe may fail reporting that not enough lines have been detected. The expected number of lines (QC.NLINE.FOUND.CLEAN) is of the order of 2550, 3950, 1300 for UVB, VIS and NIR respectively.
  - Proposed solution:
    The user should check to have accurate input calibration frames (mainly ARC_LINE_LIST_ARM, THEO_TAB_MULT_ARM (in poly mode), ORDER_TAB_EDGES_SLIT_ARM, WAVE_TAB_GUESS_ARM), but also to have a large enough value of **detectarclines-search-win-hsize** and a small enough value of the parameter **detectarclines-min-sn**.
  - Feature: Output residual tables have X/Ythanneal columns filled by NULL in case of poly mode data reduction and columns X/Ypoly filled by NULL in case of physical model mode data reduction.
  - Clarification: This because this table is initialized with all possible columns but the X/Ythanneal ones are filled only in physical model mode while X/Ypoly are filled only in poly mode.
- xsh_flexcomp:

  - Possible problem: The recipe fails reporting that the input frame is improper.
  - Proposed solution: The recipe make sure the input frame is proper, that means INS.OPTIi.NAME (i=3,4,5 for UVB, VIS, NIR) must have value "PIN_0.5".

- xsh_respon_slit_* recipes

  - Possible problem: the recipe may not generate flux calibrated spectra.
  - Possible explanation: The observed standard star is not listed in the reference catalogue.
  - Possible explanation: The user does not input the proper atmospheric exctinction table. From release 1.3 on the pipeline adopts an updated table that contains atmospheric exctinction values in a column named "EXTINCTION". The column name of the corresponding table used in previous releases was named "LA_SILLA".

- xsh_respon_slit_nod and xsh_scired_slit_nod recipes
- Possible problem: Bad quality of extracted, merged 1D spectrum (when **extract-method** set to 'NOD'). In case of xsh_respon_slit_nod also the RESPONSE_MERGE1D_SLIT_ARM would have bad quality.
- Possible solution: Set **extract-method** to 'LOCALIZATION' and use proper values for **localize-slit-position**, **localize-slit-hheight** parameters based on how the extracted, merged 2D spectrum looks like, so as to have the observed object at the center of the extraction slit and to include all positives.
- xsh_scired_slit_stare.

  - Possible problem: sky subtraction residuals present in the sky subtracted frame, particularly in NIR, due to residual tilt of the sky model with respect to observed 2D frame.
  - Proposed solution: Check carefully the accuracy of xsh_2dmap products. Run the data reduction chain in physical model mode (that provides a more robust 2D geometry solution than the poly mode), and make sure the residuals after model optimization are small, eventually executing the xsh_2dmap recipe with more iterations.
  - Proposed solution: Eventually set the parameter **sky-method** to median.
  - Proposed solution: apply flexure corrections, computed by xsh_flexcomp.
  - Possible problem: sky subtraction residuals and artifacts in UVB arm on the sky subtracted frame, using BSPLINE method.
  - Proposed solution: UVB arm has only very few sky lines. The user may have either to change the number of Bezier spline sampling points, eventually order by order by setting appropriate values in the additional input table SKY_SUB_BKPTS_ARM, or set the parameter **sky-method** to median.
  - Possible problem: Reduce frames with two object traces in the slit.
  - Proposed solution: Reduce each object trace separately by setting appropriate values for the parameters **sky-position1**, **sky-hheight1**, **sky-position2**, **sky-hheight2**. Those allow to define specific parts of the slit where the sky is to be fitted during single frame sky subtraction. By default, they are set to zero, and the sky is defined as the part outside the localization of the object (and outside the masked edges). If several objects are present along the slit or for whatever reasons, the user may want to tune these positions. Positions of the centres and half-heights of the regions are expressed in arcseconds.

- Possible problem: Artifacts in sky-subtracted image.
- Proposed solution: This is a known problem of the 'MEDIAN' sky subtraction method. Use instead 'BSPLINE' with proper number of sampling points.
- Possible problem: Sky subtraction quality not very good (some artifacts possibly appearing along the orders).
- Proposed solution: The user may need to better mask slit edges by increasing the **sky-slit-edges-mask** parameter value. Current default is 1.5 (arcsec) and may be incresed up to 3.
- Possible problem: Performance (speed and quality).
- Proposed solution: The user should use a proper value for the sampling wavelength parameter **rectify-bin-lambda** (See Table at page 91).
- Possible problem: optimal extraction in physical model mode may occasionally fail.
- Proposed solution: The user should choose proper window where to perform object localization. If extraction problem persist contact usd-help@eso.org or use standard extraction.
- Possible problem: Artefacts appearing at each 2D (and 1D) extracted order edges, and then affecting quality of merged extracted 2D (and 1D) spectra.
- Proposed solution: Use the reference format-check frames. Verify that the values of WLMIN/WLMAX are proper: they should be equal to the last wavelength imaged on the illuminated part of the order, divided by 1.007. This can be controlled also by projecting region files obtained by the test script **test_xsh_data_wave_tab_2d** onto the 2D sky subtracted science image. Alternatively, in physical model mode, the user may also provide in input a reference table with sky lines, tagged as SKY_LINE_LIST_ARM, and then project onto the 2D sky subtracted science image the predicted positions.
- Possible problem: Small jumps in order merging.
- Proposed solution: This is due to slit edges effects introduced by the flat-field. We recommend the user to use **extract-method** 'LOCALIZATION', **localize-method** 'MANUAL' with proper values of **localize-slit-position** and **localize-slit-hheight** (low S/N objects) or 'AUTO' (high S/N objects).
- Possible problem: Over-estimated error.
- Proposed solution: This is due to slit edges effects introduced by the flat-field. We recommend the user to use **extract-method** 'LOCALIZATION', **localize-method** 'MANUAL' with proper values of **localize-slit-position** and **localize-slit-hheight** (low S/N objects) or 'AUTO' (high S/N objects).
- Possible problem: The recipe may fail if **extraction-method** is set to LOCALIZATION and **localise-method** is set to eithe GAUSSIAN or MAXIMUM.
- Proposed solution: This problem, that usually does not occur in xsh_respon_slit_stare that has in input STD star data, is probably due to the sometimes low S/N ratio of science data, that make difficult the detection of the object trace either via Gaussian cross-order profile or Maximum dettection method. The user may tray to change parameters **localize-slit-position** and **localize-slit-hheight** or alternatively set **extraction-method** to MANUAL.

• xsh_scired_ifu_stare:

  - Possible problem: Recipe failing reporting errors on incompatible input AFC corrected and science frames.

  - Proposed solution: The use of proper input AFC corrected model configuration table frame (optimized by xsh_2dmap), in physical model mode, or AFC corrected wavelength solution (WAVE_TAB_AFC_ARM), in poly mode is critical to obtain good accuracy in IFU cube reconstruction. The recipe checks that the physical model table, or the wavelength solution have been AFC corrected with the proper AFC frame (INS.OPTIi.NAME='Pin_0.5', i=3,4,5; OBS.ID and OBS.TARG.NAME of the AFC frame corrected input are the same as the ones of the science frame), else stops with proper error message.

    The user should provide proper input frames.
  - Possible problem: Occasionally accuracy of cube reconstruction in binned data may be worse than 0.5 bins.
  - Possible solution: We recommend the user to check the cube accuracy first with a standard star (usually available in case of IFU observations) and then to use in input the table IFU_CFG_TAB_ARM and optimize the corresponding parameters to obtain best accuracy in the cube reconstruction of the reference standard star and then use the optimized table to reduce the science object.

- xsh_scired_ifu_offset: This recipe may have the same problems as the xsh_scired_ifu_stare and we recommend corresponding solutions.

  - Possible Problem: The produced data cube is not well aligned
  - Possible solution: Increase the parameter **–localize-ifu-chunk-hsize** to obtain a higher signal to noise
  - Possible solution: For the lateral slitlets, increase the parameters **–localizeifu-slitlow(up)-edges-mask** in order to mask the slit edges which may be noisy.
  - Possible solution: Increase (or possibly decrease) the number of iterations of the shift computation with the parameter –correctifu-niter

- xsh_scired_ifu_stare_drl:

  - Possible Problem: Upon examination of the traces, the produced data cube has a significant misalignment
  - Possible solution: Run again xsh_geom_ifu with new parameters check the quality of the 'self-aligned' cube and run xsh_scired_ifu_stare_drl with the newly produced tables.
  - Possible Problem: Upon examination of the traces, the produced data cube has a residual misalignment between slitlets of less than one pixel.
  - Possible solution: Set the parameters **shift-offsettab-low(up)** to suitable values.

- xsh_scired_ifu_offset_drl: This recipe may have the same problems as the xsh_scired_ifu_stare_drl and we recommend corresponding solutions.

7 Instrument Data Description

To reduce a science frame, the following calibration frames are needed:

- Pinhole arc-lamp format-check frames (unbinned).
- Bias(es) (for UVB/VIS only).
- Dark(s) (optional for UVB/VIS).
- Pinhole continuum-lamp order definition (unbinned).
- Flat field(s) (SLIT and IFU modes).
- Multi-pinhole arc-lamp frames (unbinned).
- Standard star observations (optional for flux calibration).
- Telluric standard star observations (optional for telluric absorption correction).
- Flexure compensation frames (required for IFU, otherwise optional for a more accurate wavelength calibration).

It is also necessary to have a set of static calibration data to hand (see next Section).

7.1 Format-check arc-lamp frames

Single pinhole format-check arc-lamp frames are taken with a ThAr lamp to allow the pipeline to obtain a guess initial wavelength solution. These frames need to be taken in sequence with the order definition and multi-pinhole arc-lamp frames to ensure that they are all aligned. The data reduction expects only unbinned frames of this kind.

7.2 Bias frames

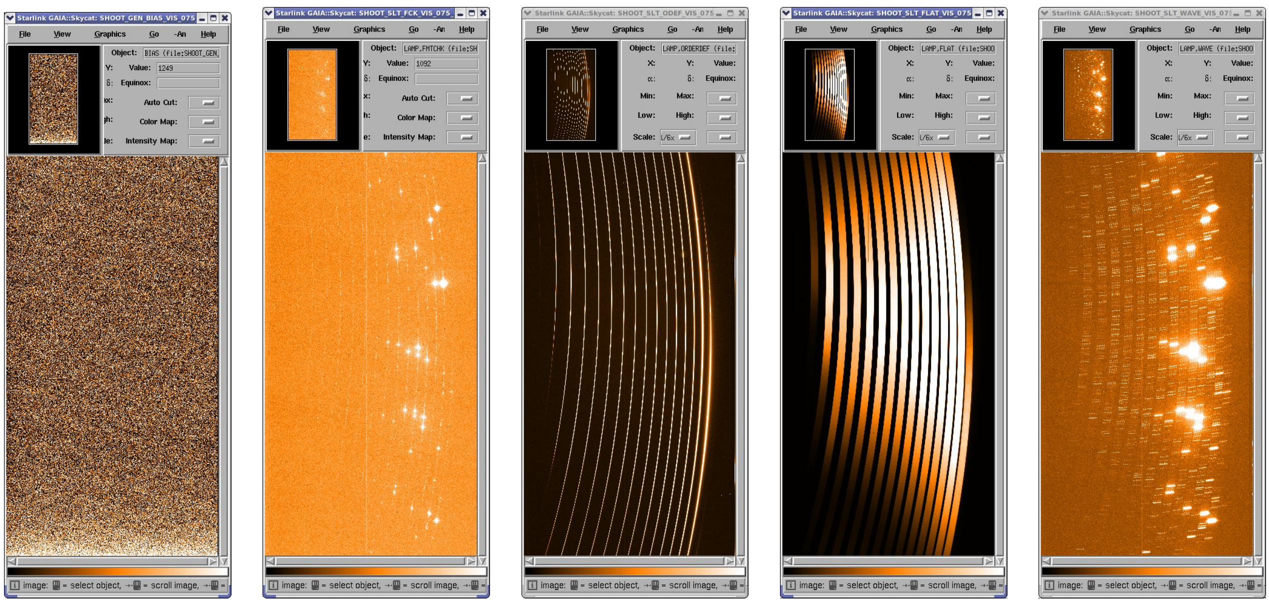
Bias frames are taken to monitor the readout noise and fixed pattern of the CCD. For XSHOOTER UVB/VIS, it is not necessary to subtract a master bias from the raw frames because there is no detectable fixed pattern noise. For these arms, the bias level is best obtained from the overscan region. For NIR, the bias level (and fixed pattern) correction is performed as part of the subtraction of the OFF observation from the ON observation.

7.3 Dark frames

For UVB/VIS, dark frames are taken occasionally in order to monitor the dark current, which is negligible. For NIR, the dark current is more significant. However dark frames in NIR are generally unnecessary because in this arm, an OFF observation is always subtracted from an ON observation, except in the case of stare observations.

7.4 Order definition frames

An order definition frame is a calibration exposure obtained with a pinhole illuminated by a continuum lamp. It is a very high signal-to-noise ratio echelle frame that precisely traces the order location. The data reduction expects only unbinned frames of this kind.



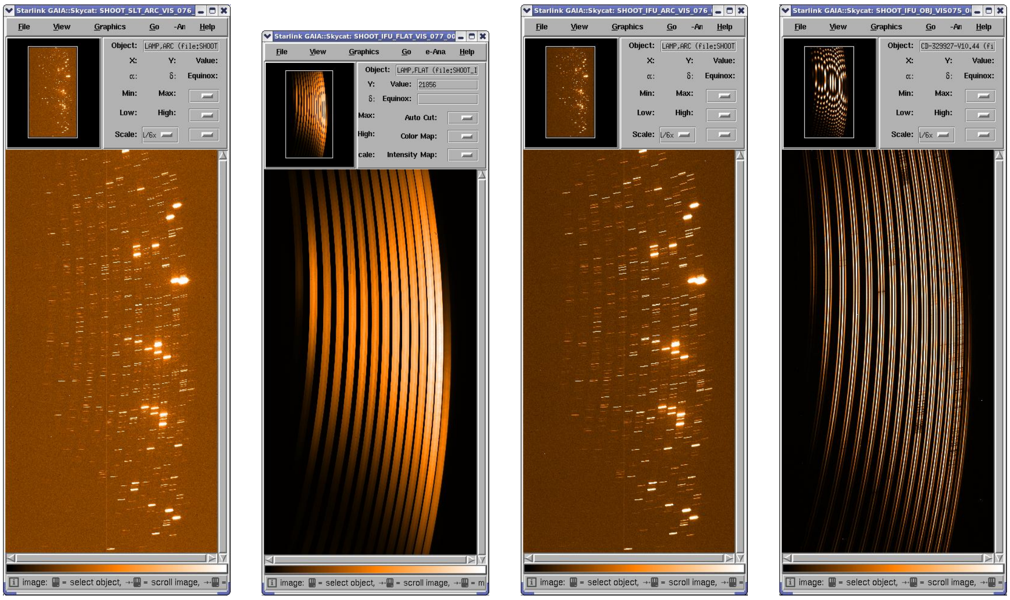
(a) (b) (c) (d) (e)

Figure 7.1: (a) a raw bias frame; (b) a raw format-check arc-lamp frame; (c) a raw order definition frame; (d) a raw flat frame; (e) a raw multi-pinhole arc-lamp frame; (VIS)

7.5 Flat Field frames

Flat frames are long slit (or IFU) exposures taken with a continuum lamp. They give information on the response of the detector, allowing to measure variations in efficiency at small (pixel-to-pixel), intermediate (fringing, in the far red) and large (the blaze function) scale. For UVB, to cover the whole wavelength range with good signal to noise, two sets of flat frames are taken, one with a halogen (QTH) lamp and the other with a deuterium ^{2}D lamp. In NIR between the J and H bands and between the H and K bands there is are absorption lines due to the light having to travel through 1m of atmosphere between the calibration lamp and the detector. In NIR the user may also have data obtained with the JH filer that suppress the signal in the K band (orders 11 and 12).

Additionally the user may want to analyse flat fields taken with the linearity template. These can be used to detect non-linear pixels. The flats come as pairs of illuminated frames (in NIR for each on frame also a corresponding off frame is acquired) taken with increasing exposure time to scan the full detector linearity range up to the non-linear regime.



(f) (g) (h) (i)

Figure 7.2: (f) a raw multi-pinhole arc-lamp frame; (g) a raw flat frame (IFU); (h) a raw multi-pinhole arc-lamp frame (IFU); (i) a raw science frame (IFU); (VIS).

7.6 Multi-pinhole arc-lamp frames

These frames are used to determine the full wavelength-spatial solution for the 2D frames. The frames are obtained by illuminating 9 pinholes with an arc-lamp. The data reduction expects only unbinned frames of this kind.

7.7 Slit arc-lamp frames

These frames are slit exposures taken with an arc-lamp and are used to compute the instrument spectral power.

7.8 IFU arc-lamp frames

IFU wavelength calibration frames are IFU exposures taken with a ThAr arc lamp. They are used to compute the instrument spectral power, and the wavelength shift(s) between the IFU slices positions and the one estimated by the multi-pinhole arc-lamp frame.

7.9 Flux standard star frames

Flux standard star calibrations are observations of stars for which the emitted spectra are known and which allows to determine the following:

- The response curve (i.e. the conversion between the non-calibrated and flux calibrated science spectrum).
- The overall efficiency of telescope+instrument+detector (corrected for atmospheric extinction). This function can be evaluated for trending.

7.10 Telluric standard star frames

The night sky spectrum from the Earth's atmosphere contains several absorption lines, namely the telluric lines. In order to have correct flux estimation the user needs to estimate this absorption by observing a so-called telluric standard star, i.e. standard stars with no absorption in the wavelength ranges where telluric absorption lines are.

7.11 Attached flexure compensation frames

Attached flexure compensation (AFC) frames are obtained by illuminating a pinhole with an arc-lamp at the current telescope pointing and they allow to compute the small shift between the daily multi-pinhole calibrations and the science observation spectral format due to changes of temperature, pressure and instrument flexure associated with the telescope pointing. Usually several AFC frames are taken. The frame that the user should use is the first one, taken with the spectrograph 0.5" pinhole (INS.OPTI2.NAME='SLOT', INS.OPTIi.NAME= 'Pin_0.5',i=3,4,5) and the same value of OBS.ID and OBS.TARG.NAME as the ones of the science frame to which it applies.

7.12 Supported raw frames

In this section we describe all possible types of raw frames for the different observing modes. More information on these data may be found at this URL. The different frame types can be identified by the values of the DPR keywords of their FITS headers (see [11]). These keywords are generated by the X-Shooter templates (for a description of the X-Shooter templates see [8]). A given frame type can be processed by one or a few different dedicated pipeline recipes. The individual pipeline recipes are described in section 11. In most cases, reference (static calibration) data are needed to reduce a given frame. These reference data have to match the input frame in a number of instrument parameters (e.g. to apply a flat field correction to a science frame only a flat field frame taken in the same arm setting, same slit width, binning, gain, read-out mode will be used for the correction). These parameters are listed under *relevant instrument parameters*.

The different kinds of raw data involved in the data reduction chain are[5] [6] [7] [8]:

Frame kind	recipe	ORIGFILE	TAG	DPR.TYPE	DPR.TECH	DPR.CATG
Linearity on	xsh_lingain [9]	SHOOT_GEN_FLAT_UVB	LINEARITY_UVB_ON	FLAT,LINEARITY,DETCHAR	IMAGE	CALIB
Linearity off	xsh_lingain	SHOOT_GEN_BIAS_UVB	LINEARITY_UVB_OFF	BIAS,LINEARITY,DETCHAR	IMAGE	CALIB
Linearity on	xsh_lingain	SHOOT_GEN_FLAT_VIS	LINEARITY_VIS_ON	FLAT,LINEARITY,DETCHAR	IMAGE	CALIB

[5] Till 30 April 2010 X-shooter file and ORIGFILE prefix is SHOOT, then XSHOO.

[6] The user can identify the kind of raw data by the DPR.TYPE, DPR.TECH, DPR.CATG values. The pipeline instead uses the frame TAG value.

[7] Flux standard stars are usually observed in on-off mode or nodding mode.

[8] In case of IFU data reduction the user may decide to use the usual recipes or the new one (xsh_geom_ifu, xsh_scired_ifu_stare, xsh_scired_ifu_offset)

[9] In the case of UVB or VIS data, the user should provide as input all linearity frames from the corresponding template (usually 32) and bias frames (usually 5). On the other hand, in the UVB and VIS, the number of non-linear pixels is negligible.

Linearity off	xsh_lingain	SHOOT_GEN_BIAS_VIS	LINEARITY_VIS_OFF	BIAS,LINEARITY,DETCHAR	IMAGE	CALIB
Linearity on	xsh_lingain	SHOOT_GEN_FLAT_NIR	LINEARITY_NIR_ON	FLAT,LINEARITY,DETCHAR	ECHELLE	CALIB
Linearity off	xsh_lingain	SHOOT_GEN_FLAT_NIR	LINEARITY_NIR_OFF	FLAT,LINEARITY,DETCHAR	IMAGE	CALIB
Bias	xsh_mbias	SHOOT_GEN_BIAS_UVB	BIAS_UVB	BIAS	IMAGE	CALIB
Bias	xsh_mbias	SHOOT_GEN_BIAS_VIS	BIAS_VIS	BIAS	IMAGE	CALIB
Dark	xsh_mdark	SHOOT_GEN_DARK_UVB	DARK_UVB	DARK	IMAGE	CALIB
Dark	xsh_mdark	SHOOT_GEN_DARK_VIS	DARK_VIS	DARK	IMAGE	CALIB
Dark	xsh_mdark	SHOOT_GEN_DARK_NIR	DARK_NIR	DARK	IMAGE	CALIB
Format-check	xsh_cfg_recover	SHOOT_SLT_FCK_UVB	FMTCHK_UVB	LAMP,FMTCHK	ECHELLE,PINHOLE	CALIB
Format-check	xsh_cfg_recover	SHOOT_SLT_FCK_VIS	FMTCHK_VIS	LAMP,FMTCHK	ECHELLE,PINHOLE	CALIB
Format-check on	xsh_cfg_recover	SHOOT_SLT_FCK_NIR_ON	FMTCHK_NIR	LAMP,FMTCHK	ECHELLE,PINHOLE	CALIB
Format-check off	xsh_cfg_recover	SHOOT_SLT_FCK_NIR_OFF	FMTCHK_NIR	LAMP,FMTCHK	IMAGE	CALIB
Format-check	xsh_predict	SHOOT_SLT_FCK_UVB	FMTCHK_UVB	LAMP,FMTCHK	ECHELLE,PINHOLE	CALIB
Format-check	xsh_predict	SHOOT_SLT_FCK_VIS	FMTCHK_VIS	LAMP,FMTCHK	ECHELLE,PINHOLE	CALIB
Format-check on	xsh_predict	SHOOT_SLT_FCK_NIR	FMTCHK_NIR_ON	LAMP,FMTCHK	ECHELLE,PINHOLE	CALIB
Format-check off	xsh_predict	SHOOT_SLT_FCK_NIR	FMTCHK_NIR_OFF	LAMP,FMTCHK	IMAGE	CALIB
Pinhole Flat	xsh_orderpos	SHOOT_SLT_ODEF_UVB	ORDERDEF_D2_UVB	LAMP,DORDERDEF	ECHELLE,PINHOLE	CALIB
Pinhole Flat	xsh_orderpos	SHOOT_SLT_ODEF_UVB	ORDERDEF_QTH_UVB	LAMP,QORDERDEF	ECHELLE,PINHOLE	CALIB
Pinhole Flat	xsh_orderpos	SHOOT_SLT_ODEF_VIS	ORDERDEF_VIS	LAMP,ORDERDEF	ECHELLE,PINHOLE	CALIB
Pinhole Flat on	xsh_orderpos	SHOOT_SLT_ODEF_NIR	ORDERDEF_NIR_ON	LAMP,ORDERDEF	ECHELLE,PINHOLE	CALIB
Pinhole Flat off	xsh_orderpos	SHOOT_SLT_ODEF_NIR	ORDERDEF_NIR_OFF	LAMP,ORDERDEF	IMAGE	CALIB
Slit Flat	xsh_mflat	SHOOT_SLT_FLAT_UVB	FLAT_D2_SLIT_UVB	LAMP,DFLAT	ECHELLE,SLIT	CALIB
Slit Flat	xsh_mflat	SHOOT_SLT_FLAT_UVB	FLAT_QTH_SLIT_UVB	LAMP,QFLAT	ECHELLE,SLIT	CALIB
Slit Flat	xsh_mflat	SHOOT_SLT_FLAT_VIS	FLAT_SLIT_VIS	LAMP,FLAT	ECHELLE,SLIT	CALIB
Slit Flat on	xsh_mflat	SHOOT_SLT_FLAT_NIR	FLAT_SLIT_NIR_ON	LAMP,FLAT	ECHELLE,SLIT	CALIB
Slit Flat off	xsh_mflat	SHOOT_SLT_FLAT_NIR	FLAT_SLIT_NIR_OFF	LAMP,FLAT	IMAGE	CALIB
IFU Flat	xsh_mflat	SHOOT_IFU_FLAT_UVB	FLAT_D2_IFU_UVB	LAMP,DFLAT	ECHELLE,IFU	CALIB
IFU Flat	xsh_mflat	SHOOT_IFU_FLAT_UVB	FLAT_QTH_IFU_UVB	LAMP,QFLAT	ECHELLE,IFU	CALIB
IFU Flat	xsh_mflat	SHOOT_IFU_FLAT_VIS	FLAT_IFU_VIS	LAMP,FLAT	ECHELLE,IFU	CALIB
IFU Flat on	xsh_mflat	SHOOT_IFU_FLAT_NIR	FLAT_IFU_NIR_ON	LAMP,FLAT	ECHELLE,IFU	CALIB
IFU Flat off	xsh_mflat	SHOOT_IFU_FLAT_NIR	FLAT_IFU_NIR_OFF	LAMP,FLAT	IMAGE	CALIB
Multipinhole arc	xsh_2dmap	SHOOT_SLT_WAVE_UVB	WAVE_UVB	LAMP,WAVE	ECHELLE,PINHOLE	CALIB
Multipinhole arc	xsh_2dmap	SHOOT_SLT_WAVE_VIS	WAVE_VIS	LAMP,WAVE	ECHELLE,PINHOLE	CALIB
Multipinhole arc on	xsh_2dmap	SHOOT_SLT_WAVE_NIR	WAVE_NIR_ON	LAMP,WAVE	ECHELLE,PINHOLE	CALIB
Multipinhole arc off	xsh_2dmap	SHOOT_SLT_WAVE_NIR	WAVE_NIR_OFF	LAMP,WAVE	IMAGE	CALIB

Frame kind	recipe	ORIGFILE	TAG	DPR.TYPE	DPR.TECH	DPR.CATG
Slit arc	xsh_wavecal	SHOOT_SLT_ARC_UVB	ARC_SLIT_UVB	LAMP,WAVE	ECHELLE,SLIT	CALIB
Slit arc	xsh_wavecal	SHOOT_SLT_ARC_VIS	ARC_SLIT_VIS	LAMP,WAVE	ECHELLE,SLIT	CALIB
Slit arc on	xsh_wavecal	SHOOT_SLT_ARC_NIR	ARC_SLIT_NIR_ON	LAMP,WAVE	ECHELLE,SLIT	CALIB
Slit arc off	xsh_wavecal	SHOOT_SLT_ARC_NIR	ARC_SLIT_NIR_OFF	LAMP,WAVE	IMAGE	CALIB
IFU arcs	xsh_wavecal	SHOOT_IFU_ARC_UVB	ARC_IFU_UVB	LAMP,WAVE	ECHELLE,IFU	CALIB
IFU arcs	xsh_wavecal	SHOOT_IFU_ARC_VIS	ARC_IFU_VIS	LAMP,WAVE	ECHELLE,IFU	CALIB
IFU arcs on	xsh_wavecal	SHOOT_IFU_ARC_NIR	ARC_IFU_NIR_ON	LAMP,WAVE	ECHELLE,IFU	CALIB
IFU arcs off	xsh_wavecal	SHOOT_IFU_ARC_NIR	ARC_IFU_NIR_OFF	LAMP,WAVE	IMAGE	CALIB
AFC arcs	xsh_flexcomp	SHOOT_IFU_AFC_UVB	AFC_ATT_UVB	LAMP,AFC	ECHELLE	CALIB
AFC arcs	xsh_flexcomp	SHOOT_IFU_AFC_VIS	AFC_ATT_VIS	LAMP,AFC	ECHELLE	CALIB
AFC arcs	xsh_flexcomp	SHOOT_IFU_AFC_NIR	AFC_ATT_NIR	LAMP,AFC	ECHELLE	CALIB
Flux std star (slit)	xsh_respon_slit_stare	SHOOT_SLT_STD_UVB	STD_FLUX_STARE_UVB	STD	ECHELLE,SLIT,STARE	CALIB
Flux std star (slit)	xsh_respon_slit_stare	SHOOT_SLT_STD_VIS	STD_FLUX_STARE_VIS	STD	ECHELLE,SLIT,STARE	CALIB
Flux std star (slit)	xsh_respon_slit_stare	SHOOT_SLT_STD_NIR	STD_FLUX_STARE_NIR	STD	ECHELLE,SLIT,STARE	CALIB
Tell std star	xsh_scired_slit_stare	SHOOT_SLT_STD_UVB	STD_TELL_STARE_UVB	STD	ECHELLE,SLIT,STARE	CALIB
Tell std star	xsh_scired_slit_stare	SHOOT_SLT_STD_VIS	STD_TELL_STARE_VIS	STD	ECHELLE,SLIT,STARE	CALIB
Tell std star	xsh_scired_slit_stare	SHOOT_SLT_STD_NIR	STD_TELL_STARE_NIR	STD	ECHELLE,SLIT,STARE	CALIB
Tell std star (on-off mode)	xsh_scired_slit_offset	SHOOT_SLT_STD_UVB	STD_TELL_OFFSET_UVB	STD	ECHELLE,SLIT,OFFSET	CALIB
Sky frame slit (on-off mode)	xsh_scired_slit_offset	SHOOT_SLT_SKY_UVB	SKY_SLIT_UVB	SKY	ECHELLE,SLIT,OFFSET	CALIB
Tell std star (on-off mode)	xsh_scired_slit_offset	SHOOT_SLT_STD_VIS	STD_TELL_OFFSET_VIS	STD	ECHELLE,SLIT,OFFSET	CALIB
Sky frame slit (on-off mode)	xsh_scired_slit_offset	SHOOT_SLT_SKY_VIS	SKY_SLIT_VIS	SKY	ECHELLE,SLIT,OFFSET	CALIB
Tell std star (on-off mode)	xsh_scired_slit_offset	SHOOT_SLT_STD_NIR	STD_TELL_OFFSET_NIR	STD	ECHELLE,SLIT,OFFSET	CALIB
Sky frame slit (on-off mode)	xsh_scired_slit_offset	SHOOT_SLT_SKY_NIR	SKY_SLIT_NIR	SKY	ECHELLE,SLIT,OFFSET	CALIB
Tell std star	xsh_scired_slit_nod	SHOOT_SLT_STD_UVB	STD_TELL_NOD_UVB	STD	ECHELLE,SLIT,NODDING	CALIB
Tell std star	xsh_scired_slit_nod	SHOOT_SLT_STD_VIS	STD_TELL_NOD_VIS	STD	ECHELLE,SLIT,NODDING	CALIB
Tell std star	xsh_scired_slit_nod	SHOOT_SLT_STD_NIR	STD_TELL_NOD_NIR	STD	ECHELLE,SLIT,NODDING	CALIB
Tell std star ifu stare	xsh_scired_ifu_stare	SHOOT_IFU_STD_UVB	OBJECT_IFU_STARE_UVB	OBJECT	ECHELLE,IFU,STARE	CALIB
Tell std star ifu stare	xsh_scired_ifu_stare	SHOOT_IFU_STD_VIS	OBJECT_IFU_STARE_VIS	OBJECT	ECHELLE,IFU,STARE	CALIB
Tell std star ifu stare	xsh_scired_ifu_stare	SHOOT_IFU_STD_NIR	OBJECT_IFU_STARE_NIR	OBJECT	ECHELLE,IFU,STARE	CALIB
Tell std star ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_OBJ_UVB	OBJECT_IFU_OFFSET_UVB	OBJECT	ECHELLE,IFU,OFFSET	CALIB
Sky ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_SKY_UVB	SKY_IFU_UVB	OBJECT	ECHELLE,IFU,OFFSET	CALIB
Tell std star ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_OBJ_VIS	OBJECT_IFU_OFFSET_VIS	OBJECT	ECHELLE,IFU,OFFSET	CALIB
Sky ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_SKY_VIS	SKY_IFU_VIS	OBJECT	ECHELLE,IFU,OFFSET	CALIB
Tell std star ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_OBJ_NIR	OBJECT_IFU_OFFSET_NIR	OBJECT	ECHELLE,IFU,OFFSET	CALIB
Sky ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_SKY_NIR	SKY_IFU_NIR	OBJECT	ECHELLE,IFU,OFFSET	CALIB
Flux std star (on-off mode)	xsh_respon_slit_offset	SHOOT_SLT_STD_UVB	STD_FLUX_STARE_UVB	STD	ECHELLE,SLIT,OFFSET	CALIB
Sky frame slit (on-off mode)	xsh_respon_slit_offset	SHOOT_SLT_SKY_UVB	SKY_SLIT_UVB	SKY	ECHELLE,SLIT,OFFSET	CALIB
Flux std star (on-off mode)	xsh_respon_slit_offset	SHOOT_SLT_STD_VIS	STD_FLUX_STARE_VIS	STD	ECHELLE,SLIT,OFFSET	CALIB
Sky frame slit (on-off mode)	xsh_respon_slit_offset	SHOOT_SLT_SKY_VIS	SKY_SLIT_VIS	SKY	ECHELLE,SLIT,OFFSET	CALIB
Flux std star (on-off mode)	xsh_respon_slit_offset	SHOOT_SLT_STD_NIR	STD_FLUX_STARE_NIR	STD	ECHELLE,SLIT,OFFSET	CALIB
Sky frame slit (on-off mode)	xsh_respon_slit_offset	SHOOT_SLT_SKY_NIR	SKY_SLIT_NIR	SKY	ECHELLE,SLIT,OFFSET	CALIB
Flux std star	xsh_respon_slit_nod	SHOOT_SLT_STD_UVB	STD_FLUX_NOD_UVB	STD	ECHELLE,SLIT,NODDING	CALIB
Flux std star	xsh_respon_slit_nod	SHOOT_SLT_STD_VIS	STD_FLUX_NOD_VIS	STD	ECHELLE,SLIT,NODDING	CALIB
Flux std star	xsh_respon_slit_nod	SHOOT_SLT_STD_NIR	STD_FLUX_NOD_NIR	STD	ECHELLE,SLIT,OFFSET	CALIB

Frame kind	recipe	ORIGFILE	TAG	DPR.TYPE	DPR.TECH	DPR.CATG
Science slit stare	xsh_scired_slit_stare	SHOOT_SLT_OBJ_UVB	OBJECT_SLIT_STARE_UVB	OBJECT	ECHELLE,SLIT,STARE	SCIENCE
Science slit stare	xsh_scired_slit_stare	SHOOT_SLT_OBJ_VIS	OBJECT_SLIT_STARE_VIS	OBJECT	ECHELLE,SLIT,STARE	SCIENCE
Science slit stare	xsh_scired_slit_stare	SHOOT_SLT_OBJ_NIR	OBJECT_SLIT_STARE_NIR	OBJECT	ECHELLE,SLIT,STARE	SCIENCE
Science ifu stare	xsh_scired_ifu_stare	SHOOT_IFU_OBJ_UVB	OBJECT_IFU_STARE_UVB	OBJECT	ECHELLE,IFU,STARE	SCIENCE
Science ifu stare	xsh_scired_ifu_stare	SHOOT_IFU_OBJ_VIS	OBJECT_IFU_STARE_VIS	OBJECT	ECHELLE,IFU,STARE	SCIENCE
Science ifu stare	xsh_scired_ifu_stare	SHOOT_IFU_OBJ_NIR	OBJECT_IFU_STARE_NIR	OBJECT	ECHELLE,IFU,STARE	SCIENCE
Science slit nod	xsh_scired_slit_nod	SHOOT_SLT_OBJ_UVB	OBJECT_SLIT_NOD_UVB	OBJECT	ECHELLE,SLIT,NODDING	SCIENCE
Science slit nod	xsh_scired_slit_nod	SHOOT_SLT_OBJ_VIS	OBJECT_SLIT_NOD_VIS	OBJECT	ECHELLE,SLIT,NODDING	SCIENCE
Science slit nod	xsh_scired_slit_nod	SHOOT_SLT_OBJ_NIR	OBJECT_SLIT_NOD_NIR	OBJECT	ECHELLE,SLIT,NODDING	SCIENCE
Science slit offset	xsh_scired_slit_offset	SHOOT_SLT_OBJ_UVB	OBJECT_SLIT_OFFSET_UVB	OBJECT	ECHELLE,SLIT,OFFSET	SCIENCE
Science slit offset	xsh_scired_slit_offset	SHOOT_SLT_SKY_UVB	SKY_SLIT_UVB	OBJECT	ECHELLE,SLIT,OFFSET	SCIENCE
Science slit offset	xsh_scired_slit_offset	SHOOT_SLT_OBJ_VIS	OBJECT_SLIT_OFFSET_VIS	OBJECT	ECHELLE,SLIT,OFFSET	SCIENCE
Science slit offset	xsh_scired_slit_offset	SHOOT_SLT_SKY_VIS	SKY_SLIT_VIS	OBJECT	ECHELLE,SLIT,OFFSET	SCIENCE
Science slit offset	xsh_scired_slit_offset	SHOOT_SLT_OBJ_NIR	OBJECT_SLIT_OFFSET_NIR	OBJECT	ECHELLE,SLIT,OFFSET	SCIENCE
Science slit offset	xsh_scired_slit_offset	SHOOT_SLT_SKY_NIR	SKY_SLIT_NIR	OBJECT	ECHELLE,SLIT,OFFSET	SCIENCE
Science ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_OBJ_UVB	OBJECT_IFU_OFFSET_UVB	OBJECT	ECHELLE,IFU,OFFSET	SCIENCE
Science ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_SKY_UVB	SKY_IFU_UVB	OBJECT	ECHELLE,IFU,OFFSET	SCIENCE
Science ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_OBJ_VIS	OBJECT_IFU_OFFSET_VIS	OBJECT	ECHELLE,IFU,OFFSET	SCIENCE
Science ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_SKY_VIS	SKY_IFU_VIS	OBJECT	ECHELLE,IFU,OFFSET	SCIENCE
Science ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_OBJ_NIR	OBJECT_IFU_OFFSET_NIR	OBJECT	ECHELLE,IFU,OFFSET	SCIENCE
Science ifu offset	xsh_scired_ifu_offset	SHOOT_IFU_SKY_NIR	SKY_IFU_NIR	OBJECT	ECHELLE,IFU,OFFSET	SCIENCE

8 Static Calibration Data

In the following section static calibration data required for X-Shooter data reduction are listed. For each of them we indicate the corresponding value of the HIERARCH ESO PRO CATG, in short PRO.CATG, FITS keyword. We use "ARM" notation as a placeholder for "UVB", "VIS" or "NIR". This has to be used to identify the frames listed in the *Set of Frames* (see Section 5.3.2, page 27).

8.1 X-Shooter Reference Spectral Format Table

To limit the data reduction to a convenient (almost complete) sub region of full spectral coverage the user needs to provide for each arm this reference input table (PRO.CATG=SPECTRAL_FORMAT_TAB_ARM). This table has several columns to control the order tracing and extraction. We recommend the use of the spectral format table called SPECTRAL_FORMAT_TAB_ARM.fits provided in the kit release.

The user should reduce NIR-JH flat, arc slit, standard star and object data using a NIR-JH custom made spectral format table that misses orders 11 and 12.

Column	Purpose
ORDER	Absolute order number
LAMP	Flat field lamp ID (used to distinguish QTH from ^{2}D flats in UVB)
WLMIN	Minimum wavelength of the corresponding order (used in order extraction) [nm]
WLMAX	Maximum wavelength of the corresponding order (used in order extraction) [nm]
DISP_MIN	Dispersion coordinate at WLMIN [pix]
DISP_MAX	Dispersion coordinate at WLMAX [pix]
LFSR	Lower free spectral range wavelength limit [nm]
UFSR	Upper free spectral range wavelength limit [nm]
WLMINFUL	Wavelength at edge of detector (low wavelength end) [nm]
WLMAXFUL	Wavelength at edge of detector (high wavelength end) [nm]
XMIN	Pixel X coordinate at min wavelength order edge (NIR only) [pix]
YMIN	Pixel Y coordinate at min wavelength order edge (NIR only) [pix]
XMAX	Pixel X coordinate at max wavelength order edge (NIR only) [pix]
YMAX	Pixel Y coordinate at max wavelength order edge (NIR only) [pix]

8.2 Line reference table

This table contains a list of emission lines from an arc lamp. The table has a PRO.CATG of ARC_LINE_LIST_ARM and the columns in the table are described below:

Column	Description
WAVELENGTH	Wavelength of the line (nm)
NAME	Name of line (if any)
FLUX	Expected central flux (e^{-})
COMMENT	Any comment

8.3 X-Shooter model configuration table

This table provides the reference parameter values for the X-Shooter model configuration. The kit provides proper model configuration tables to be used with the data. In case of significant spectral format changes, this may need to be updated using the spectral format recovery procedure described in [3].

8.4 Short list of arc lines for xsh_cfg_recover

To allow the user to recover possible spectral format shifts, we provide a table with the X and Y positions of a short list of arc lines. The user should determine the positions of the lines on the shifted format-check frame. This is one of the inputs to xsh_cfg_recover (PRO.CATG=XSH_MEASCOORD_ARM).

8.5 Line reference table for flexure compensation corrections

This table contains only a few arc lines to allow the user to compute the flexure compensation corrections (PRO.CATG=ARC_LINE_LIST_AFC_ARM). In UVB, only the line Hg 404.6 nm (generated by a pen-ray lamp) is used. For VIS only the line Ne 633 nm is used. For NIR any line list can be used since the NIR arm has no ADC along the optical path.

8.6 Standard stars flux table

This table contains photometrically aligned model spectra spectra for a given list of standard stars in units of $erg \cdot cm^{-2} \cdot s^{-1} \cdot$ Å$^{-}1$. Its TAG should be FLUX_STD_CATALOG_ARM. Current pipeline release provides a catalogue with seven flux standards: GD71, Feige 110, GD153, LTT3218, LTT7987, BD+174708, EG274.

8.7 Atmospheric extinction table

This table provides the extinction coefficient as a function of the wavelength expressed in nm. The curve compiled by Patat et al. (A&A, 527, A91, 2011) has been replaced by a typical LBLRTM model spectrum for Paranal, which covers the range 300 nm–1099.4 nm. We interpolated across the regions of strong telluric absorption in the VIS range (585.5 nm–599.2 nm, 626.1 nm–634.9 nm, 643.8 nm–660.0 nm, 682.1 nm–709.4 nm, 712.7 nm–743.4 nm, 756.2 nm–773.1 nm, 780.1 nm–861.3 nm, 879.8 nm–1033.8 nm, $>$1050 nm). Since the model is not an actual fit to the data we added 0.03 mag (well below the measurement uncertainties) to the extinction values to get them better aligned to the FORS2 data points reported by Patat et al. (2011). Its PRO.CATG is ATMOS_EXT_ARM. In NIR, the pipeline assumes no atmospheric extinction.

8.8 Table with wavelengths to fit response

The X-shooter pipeline uses stellar model spectra as reference spectra to determine the response from flux standard star observations. In some cases the line profiles in these reference spectra deviate on a 1–2% level from the actually observed ones. Thus the ratio of reference to observed spectrum may show residuals at the places of strong lines. In order to avoid such residuals and regions of strong telluric absorption the pipeline uses predefined points along the spectrum to fit the response (RESP_FIT_POINTS_CAT_ARM). They were optimized on NODDING flux standard star data (observed regularly since May 2011), so for OFFSET flux standard star data some changes may be necessary. The user should keep in mind that the response is interpolated between the tabulated points. Larger regions without fit points (e.g. the wavelength ranges 1300 nm – 1500 nm and 1700 nm – 1980 nm in the NIR arm) are purely interpolated, which may or may not give a good approximation to the true response.

8.9 Table with telluric model spectra

The X-shooter pipeline from release 2.2 offers the possibility to apply atmospheric telluric lines in the VIS and NIR wavelength ranges. This is possible by providing in input of the response recipe a FITS table that is part of the kit delivery and has PRO.CATG set to TELL_MOD_CAT_ARM[10] (ARM=VIS or NIR). Usually corresponding response computation, when this table is provided in input of the response recipes, is more accurate (despite recipe execution time increases significantly).

8.10 Sky Lines reference table

To control sky subtraction quality in the recipes xsh_scired_slit_stare and xsh_scired_ifu_stare the user may provide an input reference line list with sky lines. This should be tagged as SKY_LINE_LIST_ARM and it has the same format as the arc lamp line reference table.

8.11 Reference table to set interpolation windows on high absorption regions

To compute a smooth response curve it is necessary to interpolate the signal around high absorption regions. Optionally the user may specify these regions in a FITS table (TAG=XSH_HIGH_ABS_WIN) that lists in different rows the starting and the ending wavelength of each region to be interpolated. This can be edited by the user to control the quality of the response and flux calibrated spectra. Large masked regions (e.g. the wavelength ranges 1300 nm – 1500 nm and 1700 nm – 1980 nm in the NIR arm) are purely interpolated, which may or may not give a good approximation to the true response.

8.12 Reference bad pixel map

The kit release provides reference bad pixel maps for each arm which map the locations of known bad pixels (e.g. bad columns). The TAG is BP_MAP_RP_ARM. The pixel code associated with the reference bad pixel map is 128 (see 11.1).

8.13 Reference Table to control IFU reconstructed cube quality

As described in section 11.32 IFU cube reconstruction accuracy depends on the value of a few coefficients that affect the wavelength dependency of the spatial coordinates in the satellites IFU slices with respect to the central one (see equations 5, 6). We recommend users to always check accuracy of cube reconstruction monitoring the IFU object traces overall accuracy in the product TRACES_OBJ_IFU_ARM, for a standard star observation that is usually provided together with the object observation in IFU mode. If the accuracy of trace overlap is worse than 0.5 bins, the user has the possibility to overwrite the hard coded values of the coefficients of equations 5, 6 by setting appropriated values in the table IFU_CFG_TAB_ARM provided in the kit distribution optimizing the accuracy of IFU cube reconstruction with the standard star observation and then using the optimized table with the science observation.

[10]The default telluric catalog for the NIR may provide insufficient correction for data taken during nights with high precipitable water vapour (PWV) content. A catalog extending to higher PWV values is available at **www.eso.org/pipelines**

9 Data Reduction

9.1 Data reduction overview

In order to fully reduce a set of X-Shooter data, including calibration and science frames, the following data reduction problems need to be solved:

- **If a proper model configuration file is not available**, then the instrument model configuration parameter values that accurately describe the X-Shooter spectral format need to be determined.
- If reducing data in poly mode, then tables with predicted pinhole locations need to be built.
- The instrument spectral format needs to be determined.
- The detector bias and dark current levels should be measured and subtracted.
- The echelle orders need to be traced in a robust manner.
- The detector pixel-to-pixel sensitivity variations, and the blaze function, need to be measured and corrected for.
- A 2D transformation to rectify the X-Shooter spectral format should be derived and used to wavelength-calibrate the spectra.
- The wavelength shift between the solution obtained on the multi-pinhole frame and the slit/ifu-arc frames needs to be corrected for.
- The telescope+instrument+detector system efficiency and response should be determined.
- Instrument flexures, particularly relevant for IFU cube reconstruction quality, should be determined.
- The science data acquired either in stare, offset or nodding modes, with a slit or an IFU, should be corrected and calibrated for all of the above effects, including also cosmic ray removal and sky subtraction, 2D and 1D extraction, order merging, and, in the case of IFU data, reconstruction of a 3D data cube.

9.2 Required input data

To be able to reduce science data one needs to use raw, product data and recipes in a given sequence which provides all the necessary input to each pipeline recipe. We call this sequence data reduction cascade. The X-Shooter pipeline involves the raw input data listed in Section 7, the recipe output data listed in Section 10, and the static calibration data listed in Section 8.

9.3 Reduction cascade

The X-Shooter data reduction follows the sequence shown in the `Reflex` workflow (see Figure 9.1). See also Section 5.4 for a written description of the cascade.

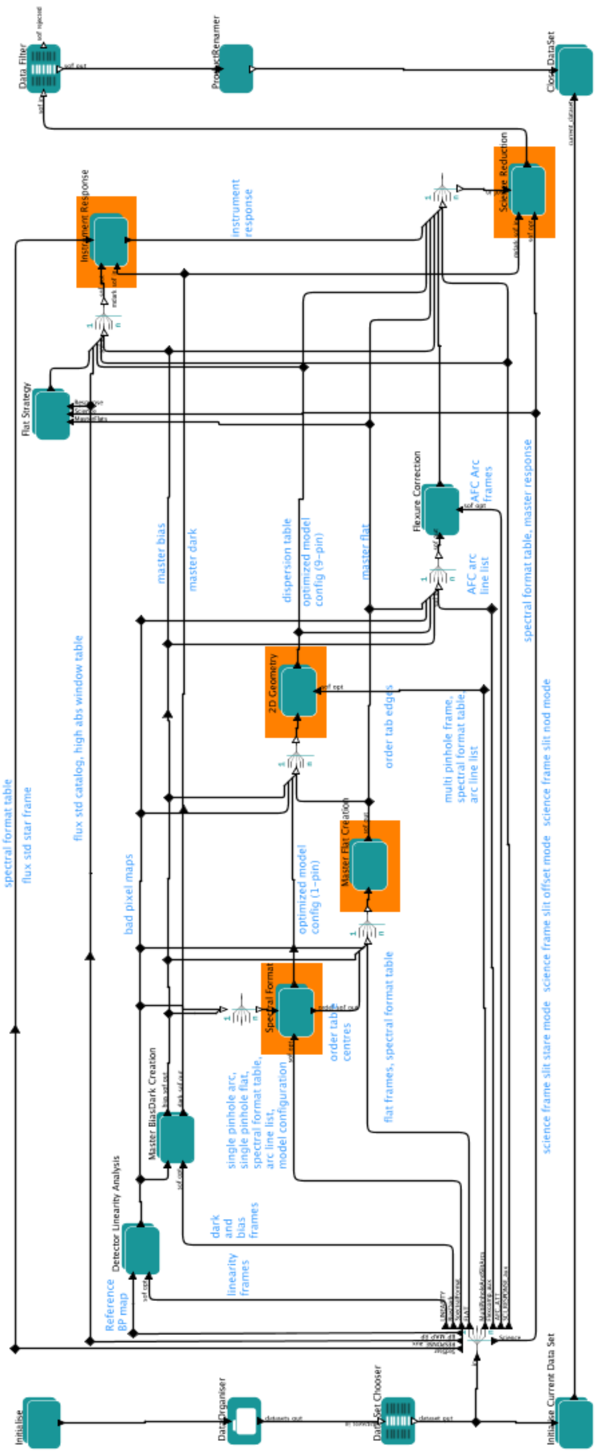


Figure 9.1: *The X-Shooter calibration cascade (physical mode).*

10 Pipeline Recipe Interfaces

In this Section we provide tables for each recipe which list the required input data and the output products, along with their classification tags. Each image product has three extensions. The 1st extension contains the pixel values, the 2nd extension contains the pixel uncertainties, and the 3rd extension contains the bad pixel map. The column "type" indicates what kind of data the user has:

- raw, raw frames.
- ref, reference frames needed by some data reduction recipe.
- cdb, pipeline products involved in the data reduction chain.
- pro, additional pipeline products.

The column "n" of the input table indicates the number of required input frames using the following convention:

- 1 for a single frame,
- + for one or more input frames,
- ! for recommended input frames,
- ? for input frames which are optional.

The column "PRO.CATG" in the output table lists the header keyword used to classify each product and to associate the correct calibration frame to each raw frame. Optional products are flagged with a "*" sign in the ID column.

In the following we use general names for files, products and frame TAG values, where the string "ARM" may be replaced by "UVB", "VIS", or "NIR".

Each recipe has the following common parameters:

parameter	default	help
keep-temp	no	If 'no', temporary files are deleted.
debug-level	none	Additional xshooter debug level. One of 'none', 'low', 'medium', 'high'.
time-stamp	FALSE	Add timestamp to product file name.
decode-bp	2144337919	Integer representation of the bits to be considered bad when decoding the bad pixel mask values.

The parameter **decode-bp** is a sum of bit-codes, each equal to a power of 2 (2^n,$n = 0, ..., 31$). The meaning of the bit-codes are described in Table 11.1. If the user wants to consider pixels with certain flags set as bad pixels, then the relevant bit-codes must be contained in the value of **decode-bp**. For example, the **decode-bp** value of 41096=8+128+8192+32768 means that pixels which are flagged with any of the codes 8, 128, 8192 or 32768 will be considered as bad pixels. Setting **decode-bp**=0 means that no pixels will be considered bad, and setting **decode-bp**=2147483647=(2^{31}-1) means that a pixel is bad if at least one flag is set.

The default value of **decode-bp** is set to 2144337919=(2^{31}-1)-1048576-2097152. In other words, all pixel flags are considered bad except for those pixels flagged as having an "extrapolated flux" or that are zero or negative in the raw data (see Table 11.1).

A full description of each recipe parameter may be obtained by running the command
esorex --params <recipe_name>, or **esorex --help <recipe_name>** or **esorex --man-page <recipe_name>**, or by looking at the Recipe Input Parameters section of the dedicated Gasgano window. Also, the role of the most relevant parameters is described in Section 11.

Quality control measurements are listed and briefly described. These are stored in the corresponding pipeline products. More information on instrument quality control can be found at `www.eso.org/qc`.

In the case of pipeline products of science recipes we use a common prefix ("PREF_") for each of the product filenames and PRO.CATGs. This is set via the following schema. "PREF"="OBJECT"_"MODE", where OBJECT is set to "FLUX" in the case of flux standard observations, "TELL" in the case of telluric standard observations, "OBJ" in the case of other science observations, and MODE is set to "SLIT", or "IFU" as appropriate.

In order to keep to a minimum the number of recipe parameters and still be able to support in a robust manner data reduction of three different arms each having different features, we have adopted the following solution. The recipes have parameter defaults that allow robust data reduction. In a few cases, some critical parameters need to have different default values for different arms. If the user has not set these critical parameters, then the recipe automatically sets appropriate arm-dependent values. Such parameters and their values are indicated for each recipe in the corresponding parameters section. To help the user to identify them on command line, if they have a numerical value, then they are defaulted to -1, and if they have a string value, then they are defaulted to 'auto'. If the user wants to perform the data reduction with a different parameter value, then the recipe employs the user choice. We recommend that the user first runs the data reduction using the parameter defaults. Then, when he/she has acquired more familiarity with the data reduction, eventually try to change parameters, starting with the critical ones, and using the quality control checks to decide on the value that is optimal for the user's data. In the case of numerical value parameters, if the user sets a value outside the indicated min/max values, then the recipes automatically set the parameter defaults.

10.1 General Recommendations

- The user should use the same spectral format table (SPECTRAL_FORMAT_TAB_ARM) through all data reduction steps. This may be a product of xsh_util_physmod, or a reference table. We provide a reference table as part of the com4 calibration data, and the corresponding instrument model configuration file, and we recommend that the user employs the reference table provided in the pipeline kit.

- To reduce IFU data, the user must improve relevant pipeline products (model configuration in physical model mode, 2D mapping in poly mode) to correct for flexure.

10.2 xsh_util_physmod

This recipe is used to generate theoretical maps containing the instrument spectral format layout corresponding to a given model configuration and list of reference arc lines, to be later on used in poly mode data reduction.

10.2.1 Input

UVB,VIS,NIR				
type	TAG	n	bin	RO
ref	XSH_MOD_CFG_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-

The input files are described in Sections 8.2 and 8.3.

10.2.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	THEO_TAB_MULT_ARM	cdb	Pinhole positions for poly mode
1	THEO_TAB_IFU_ARM	cdb	Pinhole positions for poly mode
2	THEO_TAB_SING_ARM	cdb	Pinhole positions for poly mode
3*	SPECTRAL_FORMAT_TAB_ARM	pro	Spectral format table
4*	WAVE_MAP_ARM	pro	Wave map
5*	SLIT_MAP_ARM	pro	Slit map

The THEO_TAB_SING_ARM is used to process frames with a single trace corresponding to the central pinhole (format-check, orderdef) and slit-arcs. The THEO_TAB_MULT_ARM is used to process frames with all multi-pinhole traces. The THEO_TAB_IFU_ARM is used to process IFU arc frames. The SPECTRAL_FORMAT_TAB_ARM is an additional product. We recommend the user to use the reference one provided in the kit. The last two products are generated if **wavemap** is set to TRUE. These are used later in rectification and wavelength calibration steps in the science data reduction recipes. The WAVE_MAP_ARM is a FITS image in which each pixel contains the value of the wavelength that would arrive at the centre of that pixel (for the centre of the slit). The SLIT_MAP_ARM is a FITS image in which each pixel contains the value of the slit position that would arrive at the centre of that pixel.

Theoretical table (THE) This table is provided by the X-shooter physical model and gives the expected position on the detector (X, Y) for positions in the space (λ, n, s) corresponding to arc lines from single or multi-pinhole exposures.

Column	Description	Type	Unit
Wavelength	Central wavelength of the line	float	nm
Order	Absolute order number	integer	none
slit_index	Number of the pinhole (0–8)	integer	none
slit_position	Position on the slit	float	$''$
detector_x	Corresponding X position on the detector	double	pix
detector_y	Corresponding Y position on the detector	double	pix

10.2.3 Quality control

No quality control parameters are computed.

10.2.4 Parameters

alias	default	min	max	units
binx	1	1	2	pix
biny	1	1	2	pix
spectral-format-tab	FALSE			
wavemap	FALSE			

10.2.5 Recommendations and issues

A user performing data reduction in poly mode may verify that the accuracy of the reference physical model configuration file is good enough that the corresponding THE tables (THEO_TAB_SING_ARM, THEO_TAB_MULT_ARM) can be used in the data reduction chain. In poly mode, we recommend to perform this step before starting to execute the xsh_predict and xsh_2dmap recipes. The commands to be executed are (FITS file names correspond to PRO.CATG value and ends in '.fits'):

$TDIR/test_xsh_the_map THEO_TAB_SING_VIS.fits
ds9 FMTCHK_ON_VIS.fits -region THEMAP.reg
$TDIR/test_xsh_the_map THEO_TAB_MULT_VIS.fits
ds9 -region WAVE_VIS.fits THEMAP.reg

10.3 xsh_lingain

This recipe is used to find pixels that respond to different flux levels in a different way to the majority of pixels and therefore are candidates for being bad pixels. The recipe produces a pixel mask flagging these pixels with the value 32768. This reduction step is useful for information purposes but is not strictly necessary, with the UVB/VIS detectors possessing very few such pixels.

10.3.1 Input

Flat frames must be provided as pairs with the same integration time (or DIT). The number of raw frames should be a multiple of four (usually 40) for NIR and of two (usually 32) for UVB/VIS frames. For UVB/VIS, the off-frames can be any number greater than zero, usually five, and for NIR the off-frames must be the same as the number of on-frames.

UVB,VIS				
type	TAG	n	bin	RO
raw	`LINEARITY_ARM_ON`	8..n	any	100k/400k
raw	`LINEARITY_ARM_OFF`	1..n	match	match
ref	`BP_MAP_RP_ARM`	?	match	-

NIR			
type	TAG	n	bin
raw	LINEARITY_NIR_ON	8..n	1x1
raw	LINEARITY_NIR_OFF	8..n	1x1
ref	BP_MAP_RP_NIR	?	match

Note that linearity frames have been acquired regularly only after January 1st 2010. They may be retrieved from the ESO archive via the X-shooter specific form by selecting the templates:
XSHOOTER_gen_tec_UVBCCDLin, XSHOOTER_gen_tec_VISCCDLin, and
XSHOOTER_gen_tec_NIRDetLin. Note also that all templates before Period 85 (April 1st 2010) have name SHOOT_xxx in place of XSHOOTER_xxx. For this recipe the file tag above specified refers to EsoRex data reduction. We recommend the user to perform X-shooter data reduction using reflex. If the user prefers to reduce lingain data with gasgano, and uses the XSHOOT.prefs preference file delivered with the kit, the linearity frames will be classified as DETMON_LAMP_ON, DETMON_LAMP_OFF. This classification is valid only executing the recipes detmon_opt_lg or detmon_opt_ir, delivered with the detmon pipeline kit. This means the user should also install the detmon kit, available at the same URL as this pipeline. Alternatively the user may replace DETMON_LAMP_ON by LINEARITY_ARM_ON and DETMON_LAMP_OFF by LINEARITY_ARM_OFF (ARM=UVB, VIS, NIR) and run the xsh_lingain recipe of the X-shooter pipeline. If the user reduces linearity data via reflex the workflow itself takes care to classify the data properly.

10.3.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	DET_LIN_INFO_ARM	qc	Table with linearity results
1	GAIN_INFO_ARM	qc	Table with detector gain results
2	COEFFS_CUBE_ARM	pro	Data cube with polynomial fit coefficients
3	RAW_BP_MAP_NL_ARM	pro	Map with non-linear pixel locations in RAW format. Pixel values represent the degree of the polynomial term with the outlier coefficient.
4	BP_MAP_NL_ARM	pro	Map with non-linear pixel locations in PRE format

For each distinct DIT/EXPTIME value, the linearity table (DETLIN_INFO_ARM) logs:

Name	Description
DIT	Detector integration time [s]
MED	Median intensity of $[(on_1 - off_1) + (on_2 - off_2)]/2$ [ADU/s]
MEAN	Mean intensity of $[(on_1 - off_1) + (on_2 - off_2)]/2$ [ADU/s]
MED_DIT	Ratio MED / DIT [ADU/s]
MEAN_DIT	Ratio MEAN / DIT [ADU/s]
ADL	Column DIT multiplied by mean value of column MED_DIT [ADU]
ESO DET WIN1 DIT1	Actual exposure time (UVB/VIS only) [s]

For each distinct DIT value, the gain table (GAIN_INFO_ARM) logs:

Name	Description
DIT	Detector integration time [s]
MEAN_ON1	Mean intensity of on_1 [ADU]
MEAN_ON2	Mean intensity of on_2 [ADU]
MEAN_OFF1	Mean intensity of off_1 [ADU]
MEAN_OFF2	Mean intensity of off_2 [ADU]
SIG_ON_DIF	Standard deviation σ_{ondif} in $on_1 - on_2$[ADU]
SIG_OFF_DIF	Standard deviation σ_{offdif} in $off_1 - off_2$ [ADU]
GAIN	$(< on_1 > + < on_2 > - < off_1 > - < off_2 >)/(\sigma^2_{ondif} - \sigma^2_{ofdif})$
AUTOCORR	Auto-correlation coefficient
GAIN_CORR	GAIN / AUTOCORR * DIT / NDIT
ADU	$(< on_1 > + < on_2 > - < off_1 > - < off_2 >)/2$
X_FIT	Independent (X) variable used in the fit
X_FIT_CORR	Independent (X) variable used in the fit (auto-correlation active)
Y_FIT	Dependent (Y) variable used in the fit

The cube of polynomial coefficients is created only if the parameter **pix2pix** is set to TRUE (default behaviour for this recipe). It contains (**order** + 1) planes, where each pixel in the ith plane contains the value of the fitted polynomial coefficient of degree **i** for that position. The difference and auto-correlation images are only created if **intermediate** is set to TRUE (default behaviour for this recipe is FALSE). They represent intermediate steps of the auto-correlation coefficient computation.

10.3.3 Quality control

The recipe computes the following QC parameters:
QC.CONAD, the Analog to Digital conversion factor [e^-/ADU].
QC.GAIN, the detector gain [ADU/e^-].
QC.GAIN.ERR, the error associated to the gain computation [ADU/e^-].
QC.RON , the read out noise [e^-].
QC.METHOD, a parameter that indicates the method applied to determine the previous quantities (Photon Transfer Curve method or median).
QC.COUNTS.MIN, the minimum median value used in the linearity test (in a user defined region) [ADU].
QC.COUNTS.MAX, the maximum median value used in the linearity test (in a user defined region) [ADU].
QC.LIN.EFF, the effective non-linearity correction.
QC.LIN.EFF.FLUX, the flux level at which effective non-linearity correction is computed.
QC.LIN.COEFi, the value of the linearity coefficient i.
QC.LIN.COEFi.ERR, the error value of the linearity coefficient i.
QC.FPN, the detector fixed pattern noise.
QC.ERRFIT, an estimate of the error of the fit (sum of squares of distances of data points from model values).
QC.NUM.BPM, the number of bad pixels detected according to polynomial information.

10.3.4 Parameters

alias	default	min	max	units
method	PTC			
order	3			
kappa	3.0			
niter	5			
llx	-1			pix

lly	-1			pix
urx	-1			pix
ury	-1			pix
ref_level	10000			ADU
intermediate	FALSE			
autocorr	FALSE			
collapse	TRUE			
rescale	TRUE			
pix2pix	TRUE			
bpmbin	FALSE			
m	26			pix
filter	-1			ADU
n	26			pix
tolerance	0.1			pix
pafgen	FALSE			
pafname	xsh_lingain			
exts	0			
fpn_method	HISTOGRAM			
fpn_smooth	13			pix
saturation_limit	65535.0			ADU
coeffs_cube_split	FALSE			
llx1	-1			pix
lly1	-1			pix
urx1	-1			pix
ury1	-1			pix
llx2	-1			pix
lly2	-1			pix
urx2	-1			pix
ury2	-1			pix
llx3	-1			pix
lly3	-1			pix
urx3	-1			pix
ury3	-1			pix
llx4	-1			pix
lly4	-1			pix
urx4	-1			pix
ury4	-1			pix
llx5	-1			pix
lly5	-1			pix
urx5	-1			pix
ury5	-1			pix

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		uvb	vis	nir
pix2pix	TRUE	FALSE	FALSE	TRUE
bpmbin	FALSE	FALSE	FALSE	TRUE
kappa	3.0	5.0	9.0	9.0

On this recipe there is no check on min/max allowed parameter values. The most important parameters are **kappa** and **niter** controlling the kappa-sigma clipping of non-linear pixels. The user may also set an appropriate value for the parameter **saturation_limit** to ignore frames with a median intensity level above this threshold. To reduce NIR arm the parameter **autocorr** should be set to TRUE.

10.3.5 Recommendations and issues

See Section 6.

For information on the algorithms implemented in this recipe, please see the detmon documentation [17].

10.4 xsh_mbias

This recipe creates a master bias frame and is not executed for NIR.

10.4.1 Input

UVB,VIS				
type	TAG	n	bin	RO
raw	BIAS_ARM	+	any	100/400k
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

10.4.2 Output

UVB,VIS			
ID	PRO.CATG	type	Note
0	MASTER_BIAS_ARM	cdb	Master bias

10.4.3 Quality control

This recipe computes the following quality control parameters:
QC.NPIXSAT, Number of detector saturated pixels (the saturation threshold is 65535 ADU for UVB and VIS and in NIR 45000 ADU if DIT*NDIT $<$ 1.2 s else 42000 ADU).
QC.FPIXSAT, Fraction of detector saturated pixels.
QC.MBIASAVG, the average value of the master BIAS (excluding bad pixels) [ADU].
QC.MBIASMED, the median value of the master BIAS [ADU].
QC.MBIASRMS, the RMS of the master BIAS frame (excluding bad pixels) [ADU].
QC.STRUCTXi, the Slope in BIAS frame in the X direction on region i (i=1,2). The frame is collapsed in the Y direction (excluding bad pixels) and fitted by a linear expression.
QC.STRUCTYi, the BIAS Y structure (see 11.9). QC.RONi, the Read Out Noise in the user defined region i [ADU].
QC.RONi.ERR Error on QC.RONi [ADU].
QC.RON.MASTER, the Read Out Noise measured on the master frame [ADU].
QC.FPN.MASTER, the Fixed Pattern Noise measured on the master frame.

10.4.4 Parameters

alias	default	min	max	units
stack-method	median			
klow	5	0	100	
khigh	5	0	100	

Note that the default parameters are robust.

Additional recipe input parameters, used only for QC, are:

alias	default	min	max	units
ref1_llx	-1	1	sizex	pix
ref1_lly	-1	1	sizey	pix
ref1_urx	-1	1	sizex	pix
ref1_ury	-1	1	sizey	pix
ref2_llx	-1	1	sizex	pix
ref2_lly	-1	1	sizey	pix
ref2_urx	-1	1	sizex	pix
ref2_ury	-1	1	sizey	pix
ron_llx	-1	1	sizex	pix
ron_lly	-1	1	sizey	pix
ron_urx	-1	1	sizex	pix
ron_ury	-1	1	sizey	pix
ron_hsize	4	0	min(sizex,sizey)	pix
ron_nsamples	100	1	min(sizex,sizey)	
fpn_llx	-1	1	sizex	pix
fpn_lly	-1	1	sizey	pix
fpn_urx	-1	1	sizex	pix
fpn_ury	-1	1	sizey	pix
fpn_hsize	2	0	min(sizex,sizey)	pix
fpn_nsamples	100	1	min(sizex,sizey)	
struct_refx	-1	0		pix
struct_refy	-1	0		pix
random_sizex	10	1	min(sizex,sizey)	pix
random_nsamples	100	1	min(sizex,sizey)	

10.4.5 Recommendations and issues

None.

10.5 xsh_mdark

This recipe creates a master dark frame. Creation of a master dark frame is necessary for NIR, but not for UVB/VIS, since the dark current for the UVB/VIS detectors is negligible. In NIR, the master dark frame is only necessary for single-science-frame reductions.

10.5.1 Input

UVB,VIS				
type	TAG	n	bin	RO

raw	DARK_ARM	+	any	100/400k
cdb	MASTER_BIAS_ARM	?	match	match
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	DARK_NIR	3	1x1
cdb	BP_MAP_NL_NIR	?	match
ref	BP_MAP_RP_NIR	?	match

10.5.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	MASTER_DARK_ARM	cdb	Master dark
1	DARK_ON_ARM	qc	First dark frame in PRE format
1	DARK_QC_ARM	qc	Last dark frame in PRE format

10.5.3 Quality control

The recipe computes the following quality control parameters:
QC.NPIXSAT, the number of saturated pixels.
QC.FPIXSAT, the fraction of saturated pixels.
QC.CRRATE, the number of detected cosmic ray hits per surface unit (cm2) and per second.
QC.NCRH, the number of detected cosmic ray hits.
QC.NCRH.AVG, the average number of cosmic ray hits per frame.
QC.BP-MAP.PICKUP.NOISEPIX, the number of pickup noise pixels.
QC.NHPIX, the number of noisy pixels.
QC.MDARKAVG, the average value of the master dark (excluding bad pixels) [ADU].
QC.MDARKMED, the median value of the master dark (excluding bad pixels) [ADU].
QC.MDARKRMS, the RMS of the master dark frame (excluding bad pixels) [ADU].
QC.NORMFPN, the fixed pattern noise value normalized to 1s exposure.
QC.NORMFPN.ERR, the fixed pattern noise error normalized to 1s exposure.
QC.FPN, the fixed pattern noise value.
QC.FPN.ERR, the fixed pattern noise error.
QC.MDARK.CONTAM, detector contamination in a region.

10.5.4 Parameters

alias	default	min	max	units

pre-overscan-corr	1	0	6	
stack-method	median			
klow	5	0	100	
khigh	5	0	100	

Note that the default parameters are robust.

Additional recipe input parameters, used only for QC, are:

alias	default	min	max	units
bp-output	FALSE			
crh-clip-kappa	-1.0	0	20	
crh-clip-niter	3	0	100	
crh-clip-frac	0.7	0	1	
noise-clip-kappa	9.0	0	20	
noise-clip-niter	5	1	100	
noise-clip-frac	0.7	0	1	
noise-clip-diff	0.0	0	1	
noise-lower-rejection	10.0	0	100	
noise-higher-rejection	10.0	0	100	
ref1_llx	-1	1	sizex	pix
ref1_lly	-1	1	sizey	pix
ref1_urx	-1	1	sizex	pix
ref1_ury	-1	1	sizey	pix
ron_llx	-1	1	sizex	pix
ron_lly	-1	1	sizey	pix
ron_urx	-1	1	sizex	pix
ron_ury	-1	1	sizey	pix
ron_hsize	4	0	min(sizex,sizey)	pix
ron_nsamples	100	1	min(sizex,sizey)	
fpn_llx	-1	1	sizex	pix
fpn_lly	-1	1	sizey	pix
fpn_urx	-1	1	sizex	pix
fpn_ury	-1	1	sizey	pix
fpn_hsize	2	0	min(sizex,sizey)	pix
fpn_nsamples	100	1	min(sizex,sizey)	

10.5.5 Recommendations and issues

- This recipe can only process darks that have the same exposure time.
- In UVB/VIS, the master dark is normalised to an exposure time of 1 s. In NIR, the master dark has the same exposure time as the input darks.
- In NIR, the dark frames are not bias subtracted because there is an unknown bias level present. Therefore the master dark frame for NIR is **only** suitable for reducing target frames with the **same** DIT.

10.6 xsh_predict

This recipe generates guess line and order tables that predict the positions of arc-lines and order-edges on a format-check frame using a physical model of X-shooter that takes into account information on the atmospheric pressure, temperature and the instrument setting provided by the FITS header of the raw format-check frame.

10.6.1 Input

Physical model mode (recommended):

UVB,VIS				
type	TAG	n	bin	RO
raw	FMTCHK_ARM	1	1x1	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
ref	XSH_MOD_CFG_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	FMTCHK_NIR_ON	1	1x1
raw	FMTCHK_NIR_OFF	1	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
ref	XSH_MOD_CFG_TAB_NIR	1	-
cdb	BP_MAP_NL_NIR	?	match
ref	BP_MAP_RP_NIR	?	match

Poly mode:

UVB,VIS				
type	TAG	n	bin	RO
raw	FMTCHK_ARM	1	1x1	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
cdb	THEO_TAB_SING_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	FMTCHK_NIR_ON	1	1x1
raw	FMTCHK_NIR_OFF	1	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
cdb	THEO_TAB_SING_NIR	1	-
cdb	BP_MAP_NL_NIR	?	match
ref	BP_MAP_RP_NIR	?	match

10.6.2 Output

Physical model mode:

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	ORDER_TAB_GUESS_ARM	cdb	Guess order table
1	FMTCHK_RESID_TAB_LINES_ARM	cdb	Line residuals table
2	FMTCHK_ON_ARM	qc	Bias subtracted format-check frame
3	XSH_MOD_CFG_OPT_FMT_ARM	cdb	Optimized physical model configuration table

The product table ORDER_TAB_GUESS_ARM contains in its first extension polynomial coefficients of the order solution along Y at the slit centre, its upper and lower edges, the upper and lower IFU slices edges positions along the slit, and the Y polynomial degree, the start and end Y trace position: CENCOEF0, EDGUPCOEF0, EDGLOCOEFF0, SLICUPCOEF0, SLICLOCEF0, CENCOEF1, EDGUPCOEF1, EDGLOCOEFF1, SLICUPCOEF1, SLICLOCEF1, CENCOEF2, EDGUPCOEF2, EDGLOCOEFF2, SLICUPCOEF2, SLICLOCEF2, DEGY, STARTY, ENDY. The second extension logs for each traced order its fitted polynomial-model trace position.

The product table FMTCHK_RESID_TAB_LINES_ARM contains the following columns:

Wavelength	Arc-line wavelength [nm]
Order	Absolute order number
Slit_position	Trace position relative to the slit's centre [arcsec]
Slit_index	The index value of the slit (pinhole number)
Xthpre	X coordinate of the line position predicted by the model [pix]
Ythpre	Y coordinate of the line position predicted by the model [pix]
XGauss	X position of the line measured by Gaussian fitting [pix]
YGauss	Y position of the line measured by Gaussian fitting [pix]
SigmaXGauss	Fitted x-sigma of the Gaussian fit [pix]
SigmaYGauss	Fitted y-sigma of the Gaussian fit [pix]
FwhmXGauss	Fitted x-FWHM of the Gaussian fit [pix]
FwhmYGauss	Fitted y-FWHM of the Gaussian fit [pix]
NormGauss	Fitted line flux [ADU]
SN	The ratio of the central pixel value to its uncertainty
Xpoly	X position of the line given by the polynomial solution [pix; poly-mode]
Ypoly	Y position of the line given by the polynomial solution [pix; poly-mode]
Xthanneal	X position of the line given by the physical model *after* annealing [pix]
Ythanneal	Y position of the line given by the physical model *after* annealing [pix]
ResidXPoly	Difference between the measured (Gaussian fitting) and model X positions [pix; poly-mode]
ResidYPoly	Difference between the measured (Gaussian fitting) and model Y positions [pix; poly-mode]
ResidXmodel	Difference between the measured (Gaussian fitting) and updated physical model X positions [pix]
ResidYmodel	Difference between the measured (Gaussian fitting) and updated physical model Y positions [pix]
Flag	A flag indicating the line status (0 means everything is ok)

The product table XSH_MOD_CFG_OPT_FMT_ARM is an optimised physical model configuration file which enables the prediction of the line locations in single-pinhole exposures to within 0.2 pix using the physical model. The configuration can be further optimised using 9-pinhole data (refining physical parameters that act away from the slit centre) by supplying this table as input to xsh_2dmap (see Section 10.9).

Poly mode:

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	WAVE_TAB_GUESS_ARM	cdb	Guess wave table
1	ORDER_TAB_GUESS_ARM	cdb	Guess order table
2	FMTCHK_RESID_TAB_LINES_ARM	qc	Line residuals table
3	FMTCHK_ON_ARM	qc	Bias subtracted format-check frame

The product table WAVE_TAB_GUESS_ARM contains polynomial coefficients for the wavelength solution along X and Y and the values of the polynomial order: DEGSLIT, DEGORDER, DEGLAMBDA.

10.6.3 Quality control

The recipe computes the following quality control parameters:

QC.NLINE.CAT, the number of arc lines in the input catalogue (arc line list).
QC.NLINE.FOUND, the number of lines from the arc line list that are detected in the format-check frame. Note that this value can be greater than QC.NLINE.CAT because the line at the end of one order may also appear in the echelle image at the start of the next order.
QC.NLINE.CAT.CLEAN, the number of arc lines in the subset of QC.NLINE.CAT after line detection (Gaussian fit or centroid method) and clipping of S/N threshold outliers.
QC.NLINE.FOUND.CLEAN, the number of arc lines in the subset of QC.NLINE.FOUND after clipping of S/N threshold outliers.

If executed in physical model mode: QC.MODEL.RESX_MIN, QC.MODEL.RESX_MAX, QC.MODEL.RESY_MIN, and QC.MODEL.RESY_MAX, are the minimum/maximum X/Y residuals [pix] between the detected and the optimised model-predictions of the line positions.

If executed in poly mode: QC.POLY.RESX_MIN, QC.POLY.RESX_MAX, QC.POLY.RESY_MIN and QC.POLY.RESY_MAX, are the minimum/maximum X/Y residuals [pix] between the detected and polynomial model-predictions positions of the line positions.
QC.FMTCHK.POLY.DIFFXAVG, QC.FMTCHK.POLY.DIFFXMED, QC.FMTCHK.POLY.DIFFXSTD, QC.FMTCHK.POLY.DIFFYAVG, QC.FMTCHK.POLY.DIFFYMED and QC.FMTCHK.POLY.DIFFYSTD, are the mean/median/standard deviation of the residuals [pix] between the detected and polynomial model-predictions positions of the line positions.
The recipe also computes for each order the minimum, maximum, mean and median values (QC.LINE.DIF*SPEC*i, *SPEC*=AVG,MIN,MAX,MED) of the difference in Y coordinate between adjacent detected lines. Of those values also the overall minimum, maximum, and the corresponding order where they occur are logged (QC.LINE.DIF*SPEC*, *SPEC*=AVG,MIN,MAX,MED, QC.LINE.DIF*SPEC*.ORD, *SPEC*=MIN,MAX).

10.6.4 Parameters

alias	default	min	max	units	physical mode	poly mode
pre-overscan-corr	1	0	6		Y	Y
detectarclines-fit-win-hsize	6	0	60	pix	Y	Y
detectarclines-search-win-hsize	3	0	60	pix	Y	Y
detectarclines-running-median-hsize	0	0	60	pix	Y	Y
detectarclines-wavesol-deg-lambda	5	0	10		N	Y
detectarclines-wavesol-deg-order	5	0	10		N	Y
detectarclines-ordertab-deg-y	2	0	10		Y	Y
detectarclines-min-sn	5.0	0	200		Y	Y
detectarclines-find-lines-center	gaussian				Y	Y
detectarclines-clip-sigma	2.0	0	20		N	Y
detectarclines-clip-niter	10	0	200		N	Y
detectarclines-clip-frac	0.7	0	1		N	Y
model-maxit	1000	0	10000		Y	N
model-anneal-factor	1.0	0	1		Y	N
model-scenario	3	0	8		Y	N

The most important parameters are **detectarclines-fit-win-hsize** and **detectarclines-search-win-hsize** that control the line detection. Specifically, these parameters have to be small enough not to include a doublet but large enough to be able to detect/fit the line.

10.6.5 Recommendations and issues

- To assess the accuracy of the physical model fit, the user can plot the line residuals (ResidXmodel, ResidYmodel) versus wavelength in the output residual table (FMTCHK_RESID_TAB_LINES_ARM). Note that the XSHOOTER workflow in Reflex has an interactive window that automatically displays this information (and more; see Fig. 10.1).
- Bad pixel information is not used by this recipe and therefore changing **decode-bp** has no effect.

10.7 xsh_orderpos

This recipe is used to accurately trace the order centres.

10.7.1 Input

UVB				
type	TAG	n	bin	RO
raw	ORDERDEF_D2_UVB	1	1x1	400k
raw	ORDERDEF_QTH_UVB	?	1x1	400k
ref	SPECTRAL_FORMAT_TAB_UVB	1	-	-
cdb	ORDER_TAB_GUESS_UVB	1	1x1	any
cdb	MASTER_BIAS_UVB	?	match	match
cdb	MASTER_DARK_UVB	?	match	match
cdb	BP_MAP_NL_UVB	?	match	-
ref	BP_MAP_RP_UVB	?	match	-

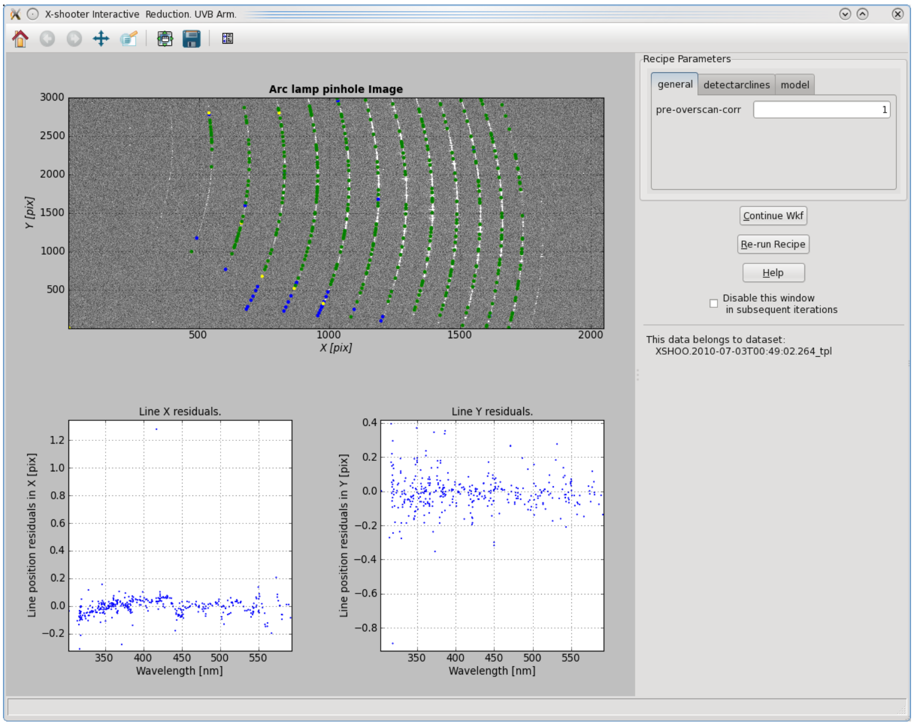


Figure 10.1: Image of the xsh_predict interactive window in Reflex.

Note that ORDERDEF_QTH_UVB frame will only be used if ORDERDEF_D2_UVB frame is not supplied.

VIS				
type	TAG	n	bin	RO
raw	ORDERDEF_VIS	1	1x1	400k
ref	SPECTRAL_FORMAT_TAB_VIS	1	-	-
cdb	ORDER_TAB_GUESS_VIS	1	1x1	any
cdb	MASTER_BIAS_VIS	?	match	match
cdb	MASTER_DARK_VIS	?	match	match
cdb	BP_MAP_NL_VIS	?	match	-
ref	BP_MAP_RP_VIS	?	match	-

NIR			
type	TAG	n	bin
raw	ORDERDEF_NIR_ON	1	1x1
raw	ORDERDEF_NIR_OFF	1	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
cdb	ORDER_TAB_GUESS_NIR	1	1x1
cdb	BP_MAP_NL_NIR	?	match
ref	BP_MAP_RP_NIR	?	match

10.7.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	`ORDER_TAB_CENTR_ARM`	cdb	Table tracing order centres
1	`ORDERPOS_RESID_TAB_ARM`	qc	Order tracing residuals table
2	`ORDERDEF_ON_ARM`	qc	Bias subtracted order frame

The product table ORDER_TAB_CENTR_ARM contains the following columns which define the detected order traces:

ORDER	Relative order number
ABSORDER	Absolute order number
CENCOEFi	Polynomial coefficients in Y for the order centroid postion (0 < i < DEGY)
EDGUPCOEFi	Polynomial coefficients in Y for the upper order edge position (0 < i < DEGY)
EDGLOCOEFi	Polynomial coefficients in Y for the lower order edge position (0 < i < DEGY)
SLICUPCOEFi	Polynomial coefficients in Y for the upper slice limit position (0 < i < DEGY)
SLICLOCOEFi	Polynomial coefficients in Y for the lower slice limit position (0 < i < DEGY)
DEGY	Degree of polynomial in Y
STARTY	Y pixel coordinate for the start of the order trace (pix)
ENDY	Y pixel coordinate for the end of the order trace (pix)

All the columns are present in the table, whatever the stage of the reduction. The undetermined coefficients are set to zero. The X position of the centroid for a given order is calculated via:

$$X = \sum_{i=0}^{n-1} Y^i \times \text{CENCOEF}_i \qquad (1)$$

The product table ORDERPOS_RESID_TAB_ARM.fits contains the residuals of the fit to the order trace with the following columns:

ORDER	Absolute order number
X	Measured X trace coordinate (pix)
Y	Y trace coordinate (pix)
RESX	Residual X trace coordinate (pix)
POLX	Predicted X trace coordinate (pix)

10.7.3 Quality control

The recipe calculates the following quality control parameters:

QC.ORD.ORDERPOS.RESIDMIN, the minimum residual in the order X positions [pix].
QC.ORD.ORDERPOS.RESIDMAX, the maximum residual in the order X positions [pix].
QC.ORD.ORDERPOS.RESIDAVG, the mean residual in the order X positions [pix].

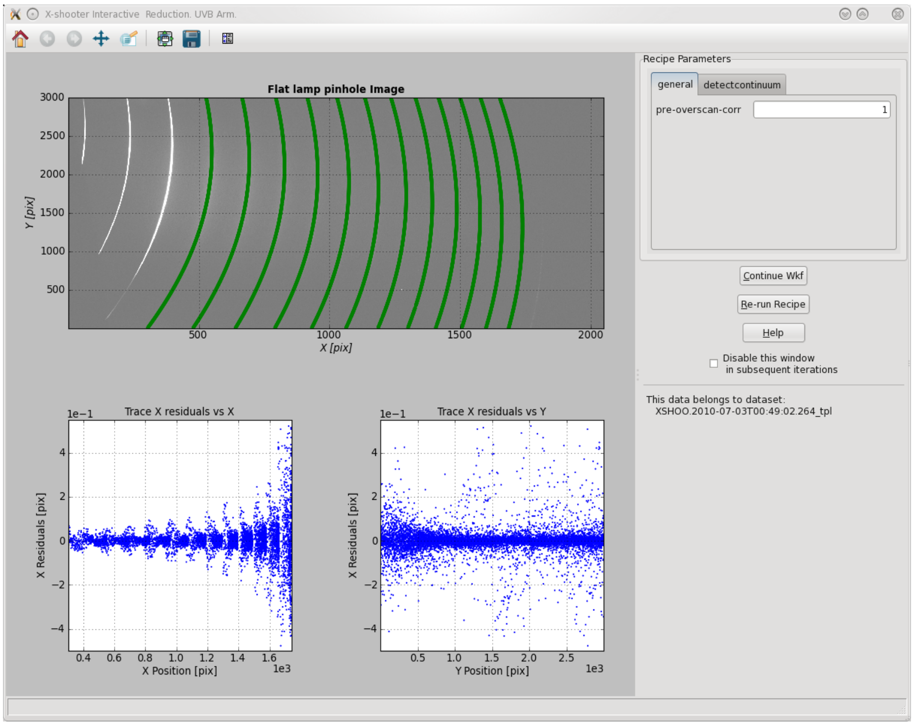


Figure 10.2: Image of the xsh_orderpos interactive window in Reflex.

QC.ORD.ORDERPOS.RESIDRMS, the RMS of the residuals in the order X positions [pix].
QC.ORD.ORDERPOS.RESELMIN, the minimum residual in the order X positions after 2σ-clipping [pix].
QC.ORD.ORDERPOS.RESELMAX, the maximum residual in the order X positions after 2σ-clipping [pix].
QC.ORD.ORDERPOS.RESELAVG, the mean residual in the order X positions after 2σ-clipping [pix].
QC.ORD.ORDERPOS.RESELRMS, the RMS of the residuals in the order X positions after 2σ-clipping [pix].
QC.ORD.ORDERPOS.NDET, the number of detected orders.
QC.ORD.ORDERPOS.NPRED, the number of orders predicted by the physical model.
QC.NPIXSAT, the number of saturated pixels.
QC.FPIXSAT, the fraction of saturated pixels.
QC.FLUXi.MIN/MAX, the minimum/maximum counts along the order trace [ADU].

10.7.4 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
detectcontinuum-search-win-hsize	5	1	100	pix
detectcontinuum-running-win-hsize	0	0	100	pix
detectcontinuum-fit-win-hsize	5	1	100	pix
detectcontinuum-center-thresh-fac	0.0	0	1	
detectcontinuum-ordertab-step-y	1	1	20	pix
detectcontinuum-ordertab-deg-y	5	0	10	
detectcontinuum-clip-res-max	0.5	0	1	pix
detectcontinuum-clip-sigma	5.0	0	20	
detectcontinuum-clip-niter	5	0	100	
detectcontinuum-clip-frac	0.4	0	1	

10.7.5 Recommendations and issues

- To assess the accuracy of the order tracing, the user can plot the order trace X residuals versus X and Y in the output residual table (ORDERPOS_RESID_TAB_ARM). Note that the XSHOOTER workflow in Reflex has an interactive window that automatically displays this information (and more; see Fig. 10.2).

10.8 xsh_mflat

This recipe creates a master flat frame, identifies hot and dead pixels, and traces the flat order edges.

10.8.1 Input

SLIT mode:

UVB				
type	TAG	n	bin	RO
raw	FLAT_D2_SLIT_UVB	+	any	100/400k
raw	FLAT_QTH_SLIT_UVB	+	any	100/400k
ref	SPECTRAL_FORMAT_TAB_UVB	1	-	-
cdb	ORDER_TAB_CENTR_UVB	1	1x1	any
cdb	MASTER_BIAS_UVB	?	match	match
cdb	MASTER_DARK_UVB	?	match	match
cdb	BP_MAP_NL_UVB	?	match	match
ref	BP_MAP_RP_UVB	?	match	match

VIS				
type	TAG	n	bin	RO
raw	FLAT_SLIT_VIS	+	any	100/400k
ref	SPECTRAL_FORMAT_TAB_VIS	1	-	-
cdb	ORDER_TAB_CENTR_VIS	1	1x1	any
cdb	MASTER_BIAS_VIS	?	match	match
cdb	MASTER_DARK_VIS	?	match	match
cdb	BP_MAP_NL_VIS	?	match	-
ref	BP_MAP_RP_VIS	?	match	-

NIR			
type	TAG	n	bin
raw	FLAT_SLIT_NIR_ON	+	1x1
raw	FLAT_SLIT_NIR_OFF	+	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
cdb	ORDER_TAB_CENTR_NIR	1	1x1
cdb	BP_MAP_NL_NIR	?	1x1
ref	BP_MAP_RP_NIR	?	1x1

In case of NIR data obtained with the JH filter the user must use an alternative spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

IFU mode:

UVB				
type	TAG	n	bin	RO
raw	FLAT_D2_IFU_UVB	+	any	100/400k
raw	FLAT_QTH_IFU_UVB	+	any	100/400k
ref	SPECTRAL_FORMAT_TAB_UVB	1	-	-
cdb	ORDER_TAB_CENTR_UVB	1	1x1	any
cdb	MASTER_BIAS_UVB	?	match	match
cdb	MASTER_DARK_UVB	?	match	match
cdb	BP_MAP_NL_UVB	?	match	match
ref	BP_MAP_RP_UVB	?	match	match

VIS				
type	TAG	n	bin	RO
raw	FLAT_IFU_ARM	+	any	100/400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	ORDER_TAB_CENTR_ARM	1	1x1	any
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	FLAT_IFU_NIR_ON	+	1x1
raw	FLAT_IFU_NIR_OFF	+	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
cdb	ORDER_TAB_CENTR_NIR	1	1x1
cdb	BP_MAP_NL_NIR	?	match
ref	BP_MAP_RP_NIR	?	match

10.8.2 Output

SLIT mode:

UVB			
ID	PRO.CATG	type	Note
0	ORDER_TAB_EDGES_QTH_SLIT_UVB	qc	Table tracing QTH master flat edges
1	MASTER_FLAT_QTH_SLIT_UVB	qc	Bias, dark, inter-order background subtracted, and normalised QTH master flat frame

2	MFLAT_QTH_BACK_SLIT_UVB	qc	QTH inter-order background frame
3	MFLAT_QTH_GRID_BACK_SLIT_UVB	qc	QTH table with background grid sampling points
4	ORDER_TAB_EDGES_D2_SLIT_UVB	qc	Table tracing D2 master flat edges
5	MASTER_FLAT_D2_SLIT_UVB	qc	Bias, dark, inter-order background subtracted, and normalised D2 master flat frame
6	MFLAT_D2_BACK_SLIT_UVB	qc	D2 inter-order background frame
7	MFLAT_D2_GRID_BACK_SLIT_UVB	qc	D2 table with background grid sampling points
8	ORDER_TAB_EDGES_SLIT_UVB	cdb	Table tracing combined master flat edges
9	MASTER_FLAT_SLIT_UVB	cdb	Bias, dark, inter-order background subtracted, and normalised combined master flat frame
10	MFLAT_BACK_SLIT_UVB	qc	Merged inter-order background frame

VIS,NIR			
ID	PRO.CATG	type	Note
0	ORDER_TAB_EDGES_SLIT_ARM	cdb	Table tracing master flat edges
1	MASTER_FLAT_SLIT_ARM	cdb	Bias, dark, inter-order background subtracted, and normalised master flat frame
2	MFLAT_BACK_SLIT_ARM	qc	Inter-order background frame
3	MFLAT_GRID_BACK_SLIT_ARM	qc	Table with background grid sampling points

IFU mode:

UVB			
ID	PRO.CATG	type	Note
0	ORDER_TAB_EDGES_QTH_IFU_UVB	qc	Table tracing QTH master flat edges
1	MASTER_FLAT_QTH_IFU_UVB	qc	Bias, dark, inter-order background subtracted, and normalised QTH master flat frame
2	MFLAT_BACK_QTH_IFU_UVB	qc	QTH inter-order background frame
3	ORDER_TAB_EDGES_D2_IFU_UVB	qc	Table tracing D2 master flat edges
4	MASTER_FLAT_D2_IFU_UVB	qc	Bias, dark, inter-order background subtracted, and normalised D2 master flat frame
5	MFLAT_BACK_D2_IFU_UVB	qc	D2 inter-order background frame
6	MFLAT_GRID_BACK_IFU_UVB	qc	Table with background grid sampling points
7	ORDER_TAB_EDGES_IFU_UVB	cdb	Table tracing combined master flat edges
8	MASTER_FLAT_IFU_UVB	cdb	Bias, dark, inter-order background subtracted, and normalised combined master flat frame
9	MFLAT_BACK_IFU_UVB	qc	Merged inter-order background frame

VIS,NIR			
ID	PRO.CATG	type	Note
0	ORDER_TAB_EDGES_IFU_ARM	cdb	Table tracing master flat edges
1	MASTER_FLAT_IFU_ARM	cdb	Bias, dark, inter-order background subtracted, and normalised master flat frame
2	MFLAT_BACK_IFU_ARM	qc	Inter-order background frame

3	MFLAT_GRID_BACK_IFU_ARM	qc	Table with background grid sampling points

The product table ORDER_TAB_EDGES_SLIT_ARM (or the equivalent product for IFU) contains three extensions. The first extension contains the same columns as those contained in the table ORDER_TAB_CENTR_ARM produced by xsh_orderpos. The second extension contains the following columns:

ORDER	Relative order number
ABSORDER	Absolute order number
CENTER_X	Predicted X coordinate of the order centre (pix)
CENTER_Y	Corresponding Y coordinate of the order centre (pix)
EDG_LO_X	Predicted X coordinate of the lower order edge position (pix)
EDG_UP_X	Predicted X coordinate of the upper order edge position (pix)
SLIC_LO_X	Predicted X coordinate of the lower slice limit position (pix)
SLIC_UP_X	Predicted X coordinate of the upper slice limit position (pix)

The third extension should be ignored.

The product table MFLAT_GRID_BACK_SLIT_ARM (or the equivalent product for IFU) has the following columns:

X	X grid position (pix)
Y	Y grid position (pix)
INT	Interorder background flux (ADU)
ERR	Uncertainty on interorder background flux (ADU)
INTfit	Fitted interorder background flux (ADU)
Residual	INT - INTfit (ADU)

10.8.3 Quality control

The recipe calculates the following quality control parameters:

QC.NPIXSAT, the number of saturated pixels.
QC.FPIXSAT, the fraction of saturated pixels.
QC.FLUXi.MIN/MAX, the minimum/maximum counts along the order [ADU].

10.8.4 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
stack-method	median			
klow	5	0	100	
khigh	5	0	100	
detectorder-edges-search-win-hsize	50	sizex/norders/4	sizex/norders	pix
detectorder-edges-flux-thresh	0.4	0	1	
detectorder-min-sn	-1.0	0	150	
detectorder-min-order-size-x	-1	1	sizex/norders	pix
detectorder-chunk-half-size	1	1	sizey	pix

detectorder-slitlet-low-factor	1.0	0	1	
detectorder-slitlet-up-factor	1.0	0	1	
detectorder-fixed-slice	TRUE			
detectorder-slice-trace-method	auto			
detectorder-qc-mode	FALSE			
detectorder-d2-min-sn	60	0	150	
background-edges-margin	1	0	15	pix
background-poly-deg-x	9	0	15	
background-poly-deg-y	9	0	15	
background-poly-kappa	10.0	0	100	

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB QTH	VIS	NIR
detectorder-min-sn	-1.0	20 (SLIT/IFU)	40 (SLIT) or 20 (IFU)	60 (SLIT) or 4 (IFU)
detectorder-min-order-size-x	-1.0	60	60	40

For IFU the value of the parameter **detectorder-slice-trace-method** is always set to “sobel”.

10.8.5 Recommendations and issues

- A polynomial model is used for the inter-order background, and it is not constrained outside of the illuminated orders. Therefore, when inspecting the master flat frame, do not worry about the poor inter-order background subtraction at the edges of the flat frame where the polynomial model diverges rapidly.
- For UVB, the master flats created for the QTH and D2 lamps are simply stitched together to create the final master flat frame MASTER_FLAT_SLIT_UVB.
- For UVB and VIS, the first flat frame in each calibration block has a different spatial form of the intra-order exposure than the remaining flat frames at the few per cent level due to the warming-up of the flat field lamps. A waiting time will be implemented in the calibration templates at some point in the near future to allow the lamp to stabilise. However, until this is implemented, the user should experiment with excluding the first flat frame in each calibration block to see if these leads to improved flat fielding, since the flat field combination algorithm in `xsh_mflat` is not appropriate for combining flat frames with different spatial forms.

10.9 xsh_2dmap

This recipe determines the two-dimensional wavelength solution needed to resample the orders.

10.9.1 Input

Physical model mode (recommended):

UVB,VIS				
type	TAG	n	bin	RO
raw	WAVE_ARM	1	1x1	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	XSH_MOD_CFG_OPT_FMT_ARM	1	1x1	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	WAVE_NIR_ON	1	1x1
raw	WAVE_NIR_OFF	1	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
cdb	ORDER_TAB_EDGES_SLIT_NIR	1	-
cdb	XSH_MOD_CFG_OPT_FMT_NIR	1	1x1
cdb	BP_MAP_NL_NIR	?	match
ref	BP_MAP_RP_NIR	?	match

Poly mode:

UVB,VIS				
type	TAG	n	bin	RO
raw	WAVE_ARM	1	1x1	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
cdb	WAVE_TAB_GUESS_ARM	1	1x1	400k
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	-	-
cdb	THEO_TAB_MULT_ARM	1	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	WAVE_NIR_ON	1	1x1
raw	WAVE_NIR_OFF	1	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
cdb	WAVE_TAB_GUESS_NIR	1	1x1
cdb	ORDER_TAB_EDGES_SLIT_NIR	1	-
cdb	THEO_TAB_MULT_NIR	1	-
cdb	BP_MAP_NL_NIR	?	match
ref	BP_MAP_RP_NIR	?	match

Note that in IFU mode the input is the same as in SLIT mode.

10.9.2 Output

Physical model mode:

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	WAVE_RESID_TAB_LINES_ARM	qc	Line residuals table
1	WAVE_MAP_ARM	qc	Wavelength map: $I(x, y) = \lambda(x, y)$
2	SLIT_MAP_ARM	qc	Slit position map
3	DISP_TAB_ARM	qc	Dispersion table
4	WAVE_ON_ARM	qc	Bias and dark subtracted pinhole frame
5	XSH_MOD_CFG_OPT_2D_ARM	cdb	Optimised cfg model

The product table WAVE_RESID_TAB_LINES_ARM contains the same columns as the product FMTCHK_RESID_TAB_LINES_ARM produced by the recipe xsh_predict.

The product table DISP_TAB_ARM contains the polynomial coefficients for the fitted dispersion relation $\lambda = \sum_{ij} c_{ij} \times T_i(X) \times T_j(Y)$ and the slit solution $s = \sum_{ij} d_{ij} \times T_i(X) \times T_j(Y)$ (where T_i is the Chebyshev polynomial of the 1st kind of degree i, and X, Y, λ and s are all normalised coordinates to the range [-1,1]). Note that we do this using a polynomial fit even though we are using a physical model because we require the inverse of the physical model transformation which converts λ to (X, Y). The columns are:

AXIS	Wavelength (“LAMBDA”) or slit (“SLIT”)
ORDER	Absolute order number
DEGX	Degree of polynomial in X
DEGY	Degree of polynomial in Y
CIJ	Coefficient ij of the polynomial

The image WAVE_MAP_ARM contains the fitted wavelength [nm] corresponding to the centre of each pixel. Similarly, the image SLIT_MAP_ARM contains the position along the slit [arcsec] relative to the slit’s centre corresponding to the centre of each pixel.

The product table XSH_MOD_CFG_OPT_FMT_ARM is an optimised physical model configuration file which enables the prediction of the line locations in nine-pinhole exposures to within 0.2 pix using the physical model. The configuration can be further optimised for a science exposure by supplying this table as input to xsh_flexcomp (see Section 10.11).

Poly mode:

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	WAVE_TAB_2D_ARM	cdb	Rectification 2D poly coeff
1	WAVE_RESID_TAB_LINES_ARM	qc	Line residuals table
2	WAVE_MAP_ARM	qc	Wavelength map: $I(x, y) = \lambda(x, y)$
3	SLIT_MAP_ARM	qc	Slit position map
4	DISP_TAB_ARM	cdb	Flux conservation dispersion table

5	WAVE_ON_ARM	qc	Bias and dark subtracted pinhole frame

The product table WAVE_TAB_2D_ARM contains the polynomial coefficients for the polynomial mode $X = \sum_{ijk} c_{ijk} \times T_i(s) \times T_j(n) \times T_k(\lambda)$ and $Y = \sum_{ijk} d_{ijk} \times T_i(s) \times T_j(n) \times T_k(\lambda)$ (where T_i is the Chebyshev polynomial of the 1st kind of degree i, and X, Y, s, n and λ are all normalised coordinates to the range [-1,1]). The columns are:

AXIS	"X" or "Y"
DEGSLIT	Degree of polynomial in s
DEGORDER	Degree of polynomial in n
DEGLAMBDA	Degree of polynomial in λ
CIJK	Coefficient ijk of the polynomial

10.9.3 Quality control

The recipe computes the following quality control parameters:

QC.NLINE.CAT, QC.NLINE.FOUND, QC.NLINE.CAT.CLEAN, QC.NLINE.FOUND.CLEAN, same as for xsh_predict.
QC.LINE.DIF*SPEC*i, *SPEC*=AVG,MIN,MAX,MED, statistics of the line position Y differences of adjacent detected lines [pix] for order i.
QC.LINE.DIF*SPEC*, *SPEC*=AVG,MIN,MAX,MED, overall statistics of QC.LINE.DIF*SPEC*i.
QC.LINE.DIF*SPEC*.ORD, the order corresponding to QC.LINE.DIF*SPEC*.

10.9.4 Parameters

alias	default	min	max	units	physical mode	poly mode
pre-overscan-corr	1	0	6		Y	Y
detectarclines-fit-win-hsize	6	0	60	pix	Y	Y
detectarclines-search-win-hsize	3	0	60	pix	Y	Y
detectarclines-running-median-hsize	0	0	60	pix	Y	Y
detectarclines-wavesol-deg-lambda	5	0	10		N	Y
detectarclines-wavesol-deg-order	4	0	10		N	Y
detectarclines-wavesol-deg-slit	1	0	10		N	Y
detectarclines-min-sn	5.0	0	200		Y	Y
detectarclines-find-lines-center	gaussian				Y	Y
detectarclines-clip-sigma	2.0	0	20		N	Y
detectarclines-clip-niter	0	0	200		N	Y
detectarclines-clip-frac	0.7	0	1		N	Y
dispersol-deg-x	4	0	10		Y	Y
dispersol-deg-y	5	0	10		Y	Y
model-maxit	500	0	10000		Y	N
model-anneal-factor	1.0	0	1		Y	N
model-scenario	4	0	8		Y	N

The most important parameters are **detectarclines-fit-win-hsize** and **detectarclines-search-win-hsize** that control the line detection. Specifically, these parameters have to be small enough not to include a doublet but large enough to be able to detect/fit the line.

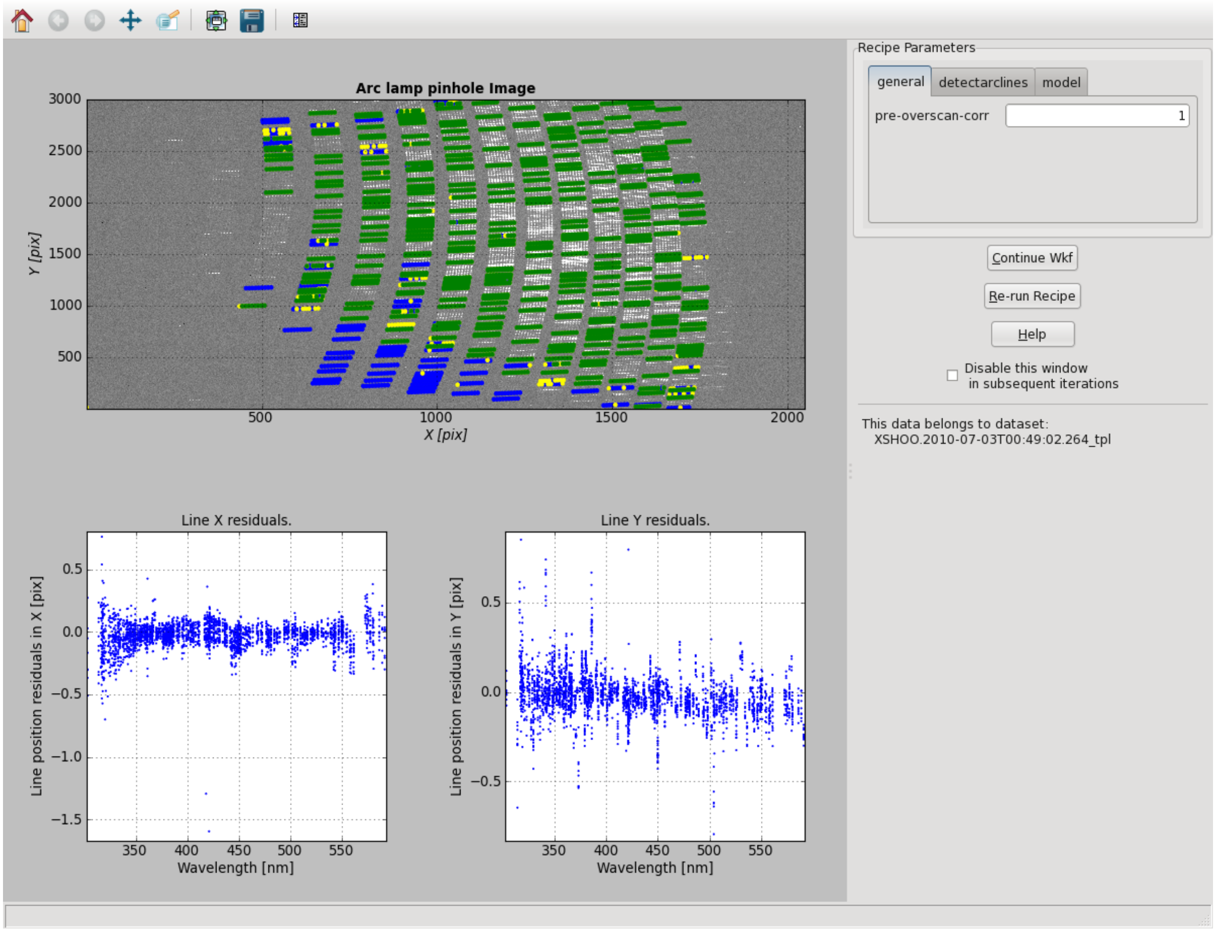


Figure 10.3: Image of the xsh_2dmap interactive window in Reflex.

10.9.5 Recommendations and issues

- To assess the accuracy of the physical model fit, the user can plot the line residuals (ResidXmodel, ResidYmodel) versus wavelength in the output residual table (WAVE_RESID_TAB_LINES_ARM). Note that the XSHOOTER workflow in Reflex has an interactive window that automatically displays this information (and more; see Fig. 10.3).
- Bad pixel information is not used by this recipe and therefore changing **decode-bp** has no effect.

10.10 xsh_wavecal

This recipe is used measure the instrument resolving power.

10.10.1 Input

Physical model mode (recommended), SLIT:

UVB,VIS				
type	TAG	n	bin	RO
raw	ARC_SLIT_ARM	1	any	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	match	match
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	XSH_MOD_CFG_OPT_2D_ARM	1	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	ARC_SLIT_NIR_ON	1	any
raw	ARC_SLIT_NIR_OFF	1	any
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
cdb	ORDER_TAB_EDGES_SLIT_NIR	1	match
cdb	XSH_MOD_CFG_OPT_2D_NIR	1	-
cdb	BP_MAP_NL_NIR	?	-
ref	BP_MAP_RP_NIR	?	-

In the case of NIR data obtained with the JH filter the user must use an alternative input spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

Poly mode, SLIT:

UVB,VIS				
type	TAG	n	bin	RO
raw	ARC_SLIT_ARM	1	any	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	DISP_TAB_ARM	1	1x1	400k
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	match	match
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	THEO_TAB_SING_ARM	1	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	ARC_SLIT_NIR_ON	1	any
raw	ARC_SLIT_NIR_OFF	1	any
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
cdb	WAVE_TAB_2D_NIR	1	1x1
cdb	DISP_TAB_NIR	1	1x1
cdb	ORDER_TAB_EDGES_SLIT_NIR	1	match
cdb	THEO_TAB_SING_NIR	1	-
cdb	BP_MAP_NL_NIR	?	-
ref	BP_MAP_RP_NIR	?	-

In the case of NIR data obtained with the JH filter the user must use an alternative input spectral format table provided in the kit release tagget as SPECTRAL_FORMAT_TAB_JH_NIR.

Physical model mode (recommended), IFU:

UVB,VIS				
type	TAG	n	bin	RO
raw	ARC_IFU_ARM	1	any	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
cdb	ORDER_TAB_EDGES_IFU_ARM	1	match	match
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	XSH_MOD_CFG_OPT_2D_ARM	1	-	-
cdb	BP_MAP_NL_UVB	?	match	-
ref	BP_MAP_RP_UVB	?	match	-

NIR			
type	TAG	n	bin
raw	ARC_IFU_NIR_ON	1	any
raw	ARC_IFU_NIR_OFF	1	any
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
cdb	ORDER_TAB_EDGES_IFU_NIR	1	match

cdb	XSH_MOD_CFG_OPT_2D_NIR	1	-
cdb	BP_MAP_NL_NIR	?	-
ref	BP_MAP_RP_NIR	?	-

Poly mode, IFU:

UVB,VIS				
type	TAG	n	bin	RO
raw	ARC_IFU_ARM	1	1x1	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_ARM	1	-	-
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	DISP_TAB_ARM	1	1x1	400k
cdb	ORDER_TAB_EDGES_IFU_ARM	1	match	match
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	THEO_TAB_IFU_ARM	1	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

NIR			
type	TAG	n	bin
raw	ARC_IFU_NIR_ON	1	1x1
raw	ARC_IFU_NIR_OFF	1	1x1
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
ref	ARC_LINE_LIST_NIR	1	-
cdb	WAVE_TAB_2D_NIR	1	1x1
cdb	DISP_TAB_NIR	1	1x1
cdb	ORDER_TAB_EDGES_IFU_NIR	1	match
cdb	THEO_TAB_IFU_NIR	1	-
cdb	BP_MAP_NL_NIR	?	-
ref	BP_MAP_RP_NIR	?	-

10.10.2 Output

SLIT:

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	TILT_TAB_SLIT_ARM	qc	Tilt and spectral resolution table
1	RESID_TAB_GOOD_LINES_ARM	qc	Line residuals table
2	ARC_ON_ARM	qc	Bias and dark subtracted arc frame
3	SHIFT_TAB_SLIT_ARM	qc	Wavelength shift table with respect to 2dmap

Note that in polynomial mode, RESID_TAB_GOOD_LINES_ARM is not produced.

IFU:

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	TILT_TAB_DOWN_IFU_ARM	qc	Tilt and spectral resolution table down slice
1	TILT_TAB_CEN_IFU_ARM	qc	Tilt and spectral resolution table central slice

2	TILT_TAB_UP_IFU_ARM	qc	Tilt and spectral resolution table up slice
3	ARC_ON_ARM	qc	Bias and dark subtracted arc IFU frame
4	SHIFT_TAB_IFU_ARM	qc	Wavelength shift table with respect to 2dmap

The product table TILT_TAB_SLIT_ARM contains the line tilt values:

ORDER	Absolute order number
WAVELENGTH	Wavelength [nm]
NAME	Empty string
CENPOSX	Line centre X position [pix]
CENPOSY	Line centre Y position [pix]
TILT	Line tilt
CHISQ	χ^2
SPECRES	Spectral resolution

Note, there is no further optimisation of the physical model configuration file by this recipe.

10.10.3 Quality control

The recipe generates the following quality control parameters:

QC.RESOLRMS, the measured RMS of resolving power of lines selected.
QC.MODEL.WAVECAL.DIFFYAVG, the average value of the differences between Y positions in the theoretical map (THE) and fitted Y positions (from the clean arc line list) [pix].
QC.MODEL.WAVECAL.DIFFYMED, the median value of the differences between Y positions in the theoretical map (THE) and fitted Y positions (from the clean arc line list) [pix].
QC.MODEL.WAVECAL.DIFFYSTD, the standard Deviation value of the differences between Y positions in the theoretical map (THE) and fitted Y positions (from the clean arc line list) [pix].
QC.NLININT, the average intensity of selected lines (at center).
QC.FWHMAVG, the mean of the FWHM measured along Y direction on selected lines [pix].
QC.FWHMRMS, the standard deviation of FWHM measured along Y direction on selected lines [pix].
QC.WAVECAL.CATLINE, the number of lines present in the catalogue.
QC.WAVECAL.FOUNDLINE, the number of lines found on the frame.
QC.WAVECAL.MATCHLINE, the number of matching lines.
QC.RESOLMED, the measured median resolving power of lines selected.
QC.RESOLRMS, the rms of the measured resolving power of lines selected.
The recipe computes for each order also the number of saturated pixels (QC.NPIXSAT) and its fraction to the total (QC.FPIXSAT).

10.10.4 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
followarclines-search-window-half-size	6	1	60	pix
followarclines-order-edges-mask	3	0	10	pix

followarclines-min-sn	-1.0	0	200	
tilt-clip-kappa	2.5	0	100	
tilt-clip-niter	5	0	100	
tilt-clip-frac	0.7	0	1	
specres-clip-kappa	2.5	0	100	
specres-clip-niter	5	0	100	
specres-clip-frac	0.7	0	1	

If the user does not set different values, then **followarclines-min-sn** is set to 15 for SLIT and 6 for IFU, for all arms.

The critical parameter is **min-sn**.

10.10.5 Recommendations and issues

We recommend that the user to verifies that the instrument spectral resolution is as expected. Figure 10.4 shows results for the three arms obtained in physical model mode. In case of 1x1 bin setting it current pipeline still under evaluate the spectral resolution in UVB and VIS. Pls refer to instrument user manual for correct values.

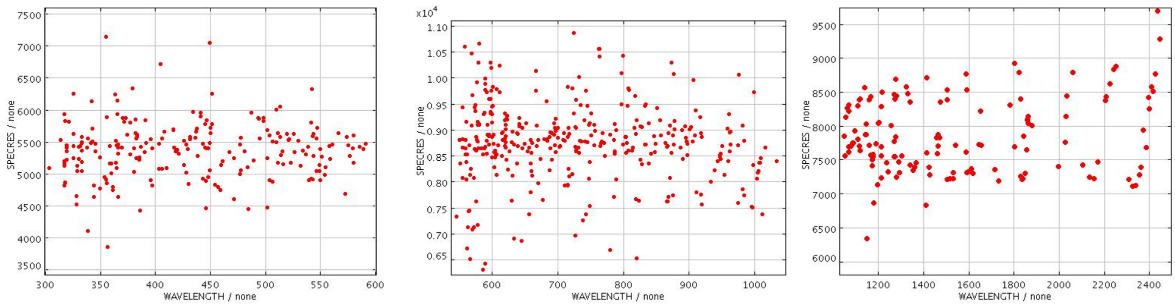


Figure 10.4: Instrument spectral resolution for the three X-shooter arms. UVB and VIS cases refer to 1x2 binning and respectively 1.0x11 and 0.9x11 slit setting, NIR to 0.6x11 slit setting. The user may compare the computed results with what reported by `ESO QC-Garching database` selecting XSHOOTER, xshooter_wavecal
.

10.11 xsh_flexcomp

This recipe is used to compute the instrument flexures.

10.11.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO

raw	AFC_ATT_ARM	1	any	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_AFC_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	ORDER_TAB_EDGES_MODE_ARM	1	-	-
ref	XSH_MOD_CFG_OPT_2D_ARM	1	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

MODE can be IFU or SLIT. The MASTER_BIAS_ARM input is only for UVB/VIS.

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	AFC_ATT_ARM	1	any	400k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
ref	ARC_LINE_LIST_AFC_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	ORDER_TAB_EDGES_MODE_ARM	1	-	-
cdb	WAVE_TAB_2D_ARM	1	1x1	400k

MODE can be IFU or SLIT. The MASTER_BIAS_ARM input is only for UVB/VIS.

10.11.2 Output

Physical model mode (recommended):

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	XSH_MOD_CFG_OPT_AFC_ARM	cdb	Flexure corrected model configuration
1	ORDER_TAB_AFC_MODE_ARM	cdb	Flexure corrected IFU traces
2	DISP_TAB_AFC_ARM	cdb	Flexure corrected dispersion table

In the physical model mode the recipe updates the values of the model configuration parameters chipx and chipy that give the location of the centre of the detector pixel array.

Poly mode:

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	WAVE_TAB_AFC_ARM	cdb	Flexure corrected wavelength solution
1	ORDER_TAB_AFC_MODE_ARM	cdb	Flexure corrected IFU traces
2	DISP_TAB_AFC_ARM	cdb	Flexure corrected dispersion table

10.11.3 Quality control

In the physical model mode, the user should check the recipe logs to make sure that the mean residuals reported by this recipe are below ∼0.05 and 0.1 pix in x and y respectively. If xsh_flexcomp is not run, then the shifts due to instrument flexure can be greater than one pixel. This can cause problems for IFU data reductions and for sky line subtraction in slit stare mode.

10.11.4 Parameters

alias	default	min	max	units	physical mode	poly mode
pre-overscan-corr	1	0	6		Y	Y
detectarclines-fit-win-hsize	6	0	60	pix	Y	Y
detectarclines-search-win-hsize	3	0	60	pix	Y	Y
detectarclines-running-median-hsize	0	0	60	pix	Y	Y
detectarclines-wavesol-deg-lambda	5	0	10		N	Y
detectarclines-wavesol-deg-order	5	0	10		N	Y
detectarclines-min-sn	5.0	0	100		Y	Y
detectarclines-find-lines-center	gaussian				Y	Y
detectarclines-clip-sigma	2.0	0	20		N	Y
detectarclines-clip-niter	10	0	200		N	Y
detectarclines-clip-frac	0.7	0	1		N	Y
dispersol-deg-x	4	0	10		Y	Y
dispersol-deg-y	5	0	10		Y	Y
model-maxit	1000	0	10000		Y	N
model-anneal-factor	1.0	0	1		Y	N
model-scenario	3	0	8		Y	N

10.11.5 Recommendations and issues

None.

10.12 xsh_respon_slit_stare

This recipe computes the instrument response and the telescope + instrument + detector efficiency.

10.12.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	slit	RO
raw	STD_FLUX_SLIT_STARE_ARM	1	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	XSH_MOD_CFG_OPT_2D/AFC_ARM	1	match	match
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	ORDER_TAB_EDGES/AFC_SLIT_ARM	1	match	match

cdb	DISP_TAB_ARM/DISP_TAB_AFC_ARM	1	match	match
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	FLUX_STD_CATALOG_ARM	1	-	-
ref	ATMOS_EXT_ARM	1	-	-
ref	SKY_SUB_BKPTS_ARM	?	-	-
ref	RESP_FIT_POINTS_CAT_ARM	1	-	-
ref	TELL_MOD_CAT_ARM	?!	-	-

For NIR there is no input master bias but a master dark with the same DIT as the input science frame is required. In case of NIR data obtained with the JH filter the user must use an alternative spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

Poly mode:

UVB,VIS,NIR				
type	TAG	n	slit	RO
raw	STD_FLUX_SLIT_STARE_ARM	1	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	match	match
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	DISP_TAB_ARM	1	1x1	400k
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	FLUX_STD_CATALOG_ARM	1	-	-
ref	ATMOS_EXT_ARM	1	-	-
ref	SKY_SUB_BKPTS_ARM	?	-	-
ref	RESP_FIT_POINTS_CAT_ARM	1	-	-
ref	TELL_MOD_CAT_ARM	?!	-	-

10.12.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	RESPONSE_SLIT_ORDER1D_ARM	cdb	Order by order instrument response table
1	RESPONSE_SLIT_MERGE1D_ARM	cdb	Merged instrument response table
2	<PREF>_ORDER2D_ARM	pro	Order by order 2D spectrum
3	<PREF>_ORDER1D_ARM	pro	Order by order 1D spectrum
4	<PREF>_MERGE2D_ARM	pro	Merged 2D spectrum
5	<PREF>_MERGE1D_ARM	pro	Merged 1D spectrum
6	<PREF>_FLUX_ORDER2D_ARM	pro	Order by order flux calibrated 2D spectrum
7	<PREF>_FLUX_ORDER1D_ARM	pro	Order by order flux calibrated 1D spectrum
8	<PREF>_FLUX_MERGE2D_ARM	pro	Merged flux calibrated 2D spectrum
9	<PREF>_FLUX_MERGE1D_ARM	pro	Merged flux calibrated 1D spectrum
10	EFFICIENCY_ARM	cdb	Telescope+instrument+detector efficiency
11	<PREF>_WAVE_MAP_ARM	pro	Wave map frame
12	<PREF>_SLIT_MAP_ARM	pro	Slit map frame

10.12.3 Quality control

This recipe generates the instrument response (RESPONSE_SLIT_ORDER1D_ARM, RESPONSE_SLIT_MERGE1D_ARM) and the instrument efficiency (EFFICIENCY_ARM) tables that are monitored by quality control.

10.12.4 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
background-edges-margin	1	0	15	pix
background-poly-deg-x	9	0	15	
background-poly-deg-y	9	0	15	
background-poly-deg-kappa	10.0	0	100	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	1000	
rectify-kernel	tanh			
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
sky-subtract	TRUE			
sky-bspline-nbkpts-first	3000	1	20000/(0.75 *biny)	
sky-bspline-nbkpts-second	3000	1	20000/(0.75 *biny)	
sky-bspline-order	7	0	12	
sky-bspline-niter	20	0	100	
sky-bspline-kappa	5.0	0	10	
sky-method	MEDIAN			
bspline-sampling	FINE			
sky-median-hsize	20	0	2000	
sky-slit-edges-mask	0.5	0	7	arcsec
sky-position1	0.0			arcsec
sky-hheight1	0.0	0	7	arcsec
sky-position2	0.0			arcsec
sky-hheight2	0.0	0	7	arcsec
stdextract-interp-hsize	30	0	1000	pix
do-optextract	FALSE			
optextract-oversample	5	0	100	
optextract-box-half-size	10	0	100	pix
optextract-chunk-size	50	0	100	pix
optextract-step-lambda	0.02	0	210	nm
optextract-clip-kappa	3.0	0	200	
optextract-clip-frac	0.4	0	1	
optextract-clip-niter	2	0	200	
optextract-niter	1	0	200	
optextract-method	GAUSSIAN			
correct-tellurics	TRUE			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.12.5 Recommendations and issues

- See the recipes xsh_scired_slit_stare and xsh_respon_slit_nod.

10.13 xsh_respon_slit_offset

This recipe computes the instrument response and the telescope + instrument + detector efficiency.

10.13.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	STD_FLUX_SLIT_OFFSET_ARM	1..N	any	100k
raw	SKY_SLIT_ARM	1..N	match	match
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	XSH_MOD_CFG_OPT_2D_ARM	1	1x1	400k
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	match	match
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM	1	1x1	400k
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	FLUX_STD_CATALOG_ARM	1	-	-
ref	ATMOS_EXT_ARM	1	-	-
ref	RESP_FIT_POINTS_CAT_ARM	1	-	-
ref	TELL_MOD_CAT_ARM	?!	-	-

In case of NIR data obtained with the JH filter the user must use an alternative spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	STD_FLUX_SLIT_OFFSET_ARM	1..N	any	100k
raw	SKY_SLIT_ARM	1..N	match	match
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	match	match

cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM	1	1x1	400k
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	FLUX_STD_CATALOG_ARM	1	-	-
ref	ATMOS_EXT_ARM	1	-	-
ref	RESP_FIT_POINTS_CAT_ARM	1	-	-
ref	TELL_MOD_CAT_ARM	?!	-	-

10.13.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	RESPONSE_ORDER1D_SLIT_ARM	cdb	Order by order instrument response table
1	RESPONSE_MERGE1D_SLIT_ARM	cdb	Merged instrument response table
2	<PREF>_ORDER2D_ARM	pro	Order by order 2D spectrum
3	<PREF>_ORDER1D_ARM	pro	Order by order 1D spectrum
4	<PREF>_MERGE2D_ARM	pro	Merged 2D spectrum
5	<PREF>_MERGE1D_ARM	pro	Merged 1D spectrum
6	<PREF>_FLUX_ORDER2D_ARM	pro	Order by order flux calibrated 2D spectrum
7	<PREF>_FLUX_ORDER1D_ARM	pro	Order by order flux calibrated 1D spectrum
8	<PREF>_FLUX_MERGE2D_ARM	pro	Merged flux calibrated 2D spectrum
9	<PREF>_FLUX_MERGE1D_ARM	pro	Merged flux calibrated 1D spectrum
10	<PREF>_SKY_ARM	pro	Sky frame
11	SKY_SLIT_ORDER2D_ARM	pro	Order by order 2D sky spectrum
12	SKY_SLIT_MERGE2D_ARM	pro	Merged 2D sky spectrum
13	EFFICIENCY_SLIT_ARM	cdb	Telescope+instrument+detector efficiency
14	<PREF>_WAVE_MAP_ARM	pro	Wave map frame
15	<PREF>_SLIT_MAP_ARM	pro	Slit map frame

10.13.3 Quality control

This recipes generate the instrument response (RESPONSE_SLIT_ORDER1D_ARM, RESPONSE_SLIT_MERGE1D_ARM) and the instrument efficiency (EFFICIENCY_ARM) tables that are monitored from quality control.

10.13.4 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	1000	
rectify-kernel	tanh			
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec

localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
stdextract-interp-hsize	30	0	1000	pix
combinenod-method	MEAN			
gen-sky	TRUE			
correct-tellurics	TRUE			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.13.5 Recommendations and issues

- See the recipes xsh_scired_slit_offset and xsh_respon_slit_nod.
- OFFSET data often do not provide a good sky correction for the NIR arm. In such cases the user can try instead to process the STD,FLUX frame only with xsh_respon_slit_stare, where the sky is fit on the frame itself.

10.14 xsh_respon_slit_nod

This recipe computes the instrument response and the telescope + instrument + detector efficiency.

10.14.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	STD_FLUX_SLIT_NOD_ARM	1..N	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	XSH_MOD_CFG_OPT_2D_ARM	1	1x1	400k
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	match	match
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM	1	1x1	400k
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-

ref	FLUX_STD_CATALOG_ARM	1	-	-
ref	ATMOS_EXT_ARM	1	-	-
ref	RESP_FIT_POINTS_CAT_ARM	1	-	-
ref	TELL_MOD_CAT_ARM	?!	-	-

In case of NIR data obtained with the JH filter the user must use an alternative spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	STD_FLUX_SLIT_NOD_ARM	1..N	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	ORDER_TAB_EDGES_SLIT_ARM	1	match	match
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM	1	1x1	400k
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	FLUX_STD_CATALOG_ARM	1	-	-
ref	ATMOS_EXT_ARM	1	-	-
ref	RESP_FIT_POINTS_CAT_ARM	1	-	-
ref	TELL_MOD_CAT_ARM	?!	-	-

10.14.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	RESPONSE_ORDER1D_SLIT_ARM	cdb	Order by order instrument response table
1	RESPONSE_MERGE1D_SLIT_ARM	cdb	Merged instrument response table
2	<PREF>_ORDER2D_ARM	pro	Order by order 2D spectrum
3	<PREF>_ORDER1D_ARM	pro	Order by order 1D spectrum
4	<PREF>_MERGE2D_ARM	pro	Merged 2D spectrum
5	<PREF>_MERGE1D_ARM	pro	Merged 1D spectrum
6	<PREF>_WAVE_MAP_ARM	pro	Wave map frame
7	<PREF>_SLIT_MAP_ARM	pro	Slit map frame
8	<PREF>_FLUX_ORDER2D_ARM	pro	Order by order flux calibrated 2D spectrum
9	<PREF>_FLUX_ORDER1D_ARM	pro	Order by order flux calibrated 1D spectrum
10	<PREF>_FLUX_MERGE2D_ARM	pro	Merged flux calibrated 2D spectrum
11	<PREF>_FLUX_MERGE1D_ARM	pro	Merged flux calibrated 1D spectrum
12	EFFICIENCY_ARM	cdb	Telescope+instrument+detector efficiency

10.14.3 Quality control

This recipes generate the instrument response (RESPONSE_SLIT_ORDER1D_ARM, RESPONSE_SLIT_MERGE1D_ARM) and the instrument efficiency (EFFICIENCY_ARM) tables that are monitored from quality control.

10.14.4 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
stack-method	median			
klow	5.0	0	100	
khigh	5.0	0	100	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	1000	
rectify-kernel	tanh			
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
rectify-fast	TRUE			
localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
localize-nod-throw	0.0			
extract-method	NOD			
stdextract-interp-hsize	30	0	1000	
combinenod-throwlist	throwlist.asc			
combinenod-method	MEAN			
max-slit	5.7			arcsec
min-slit	-5.3			arcsec
correct-tellurics	TRUE			
correct-sky-by-median	TRUE			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.14.5 Recommendations and issues

- See the recipe xsh_scired_slit_nod.
- The user may want to compare the flux calibrated 1D spectrum of the observed spectrum with the one contained in the reference catalogue (FLUX_STD_CATALOG_ARM). Another check may be to cross calibrate one flux STD star with another.
- The RESPONSE_SLIT_MERGE1D_ARM product contains in addition to the columns LAMBDA, RESPONSE, to log at each sampling wavelength the corresponding fitted instrument response, also the columns OBS, and REF, and REV_DIV_OBS. OBS contains the observed standard star spectrum (MERGE1D), corrected for exposure time, gain, atmospheric extinction and (for VIS and NIR data) telluric absorption. REF contains the reference star spectrum from the reference catalogue, but aligned to the same radial

velocity as the observed standard star spectrum. REF_DIV_OBS contains the ratio of REF and OBS and can be compared directly to the fitted response. Editing the file RESP_FIT_POINTS_CAT_ARM allows the user to change the wavelength points used to fit the response. For data with well corrected telluric absorption more fitting points may be added to fit a larger part of the response.

10.15 xsh_scired_slit_stare

This recipe reduces the science frames observed in slit stare mode.

10.15.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_SLIT_STARE_ARM	1..N	any	any
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	XSH_MOD_CFG_OPT_2D/AFC_ARM	1	1x1	400k
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	ORDER_TAB_EDGES/AFC_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM/DISP_TAB_AFC_ARM	?	1x1	400k
cdb	RESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	MRESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	ATMOS_EXT_ARM	?	-	-
ref	SKY_SUB_BKPTS_ARM	?	-	-
ref	SKY_LINE_LIST_ARM	?	-	-

In case of NIR data obtained with the JH filter the user must use an alternative spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

For NIR there is no input master bias but a master dark with the same DIT as the input science frame is required.

All OBJECT_SLIT_STARE_ARM science frames must have the same exposure time.

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_SLIT_STARE_ARM	1..N	any	any
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	WAVE_TAB_2D_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_DARK_ARM	?	match	match
cdb	MASTER_FLAT_SLIT_ARM	1	match	match

cdb	ORDER_TAB_EDGES/AFC_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM/DISP_TAB_AFC_ARM	1	1x1	400k
cdb	RESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	MRESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	ATMOS_EXT_ARM	?	-	-
ref	SKY_SUB_BKPTS_ARM	?	-	-

10.15.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	<PREF>_ORDER2D_ARM	pro	Order by order 2D spectrum
1	<PREF>_ORDER1D_ARM	pro	Order by order 1D spectrum
2	<PREF>_MERGE2D_ARM	pro	Merged 2D spectrum
3	<PREF>_MERGE1D_ARM	pro	Merged 1D spectrum
4	<PREF>_SUB_BACK_SLIT_ARM	qc	Science frame - corrected for bias level, master dark, inter-order background
5	<PREF>_WAVE_MAP_ARM	qc	Wavelength map
6	<PREF>_SLIT_MAP_ARM	qc	Slit position map
7	<PREF>_DIVFF_ARM	qc	Science frame - corrected for bias level, master dark, inter-order background, sky subtracted, flat fielded
8	<PREF>_SUB_SKY_ARM	qc	Science frame - corrected for bias level, master dark, inter-order backgound, sky subtracted
9	<PREF>_SKY_ARM	qc	Sky frame
10	<PREF>_SKY_ORD1D_ARM	qc	Order by order 1D sky spectrum
11	<PREF>_BACK_SLIT_ARM	qc	Inter-order background frame
12	<PREF>_ON_ARM	qc	Science frame in PRE format
13	<PREF>_FLUX_ORDER2D_ARM	pro	Order by order flux calibrated 2D spectrum
14	<PREF>_FLUX_ORDER1D_ARM	pro	Order by order flux calibrated 1D spectrum
15	<PREF>_FLUX_MERGE2D_ARM	pro	Merged flux calibrated 2D spectrum
16	<PREF>_FLUX_MERGE1D_ARM	pro	Merged flux calibrated 1D spectrum
17	SKY_SLIT_MERGE2D_ARM	qc	Merged 2D sky spectrum
18	SKY_TAB_MULT_ARM	qc	Predicted 9-pinhole positions of the sky lines

The flux calibrated spectrum products are only written out if a response curve is supplied to the recipe.

10.15.3 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
stack-method	median			
klow	5.0	0	100	
khigh	5.0	0	100	
background-edges-margin	1	0	15	pix
background-poly-deg-x	9	0	15	

background-poly-deg-y	9	0	15	
background-poly-deg-kappa	10.0	0	100	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	1000	
rectify-kernel	tanh			
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
sky-subtract	TRUE			
sky-bspline-nbkpts-first	3000	1	20000/(0.75 *biny)	
sky-bspline-nbkpts-second	3000	1	20000/(0.75 *biny)	
sky-bspline-order	7	0	12	
sky-bspline-niter	20	0	100	
sky-bspline-kappa	5.0	0	10	
sky-method	MEDIAN			
bspline-sampling	FINE			
sky-median-hsize	20	0	2000	
sky-slit-edges-mask	0.5	0	7	arcsec
sky-position1	0.0			arcsec
sky-hheight1	0.0	0	7	arcsec
sky-position2	0.0			arcsec
sky-hheight2	0.0	0	7	arcsec
stdextract-interp-hsize	30	0	1000	pix
do-optextract	FALSE			
optextract-oversample	5	0	100	
optextract-box-half-size	10	0	100	pix
optextract-chunk-size	50	0	100	pix
optextract-step-lambda	0.02	0	210	nm
optextract-clip-kappa	3.0	0	200	
optextract-clip-frac	0.4	0	1	
optextract-clip-niter	2	0	200	
optextract-niter	1	0	200	
optextract-method	GAUSSIAN			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.15.4 Recommendations and issues

- The input science frames must all have the same exposure time and they are assumed to have been observed such that the object is at the same position on the slit in all frames. If this is not the case, then the recipe reduction cascade is not appropriate, and the science frames should be reduced by this recipe individually.

- The threshold for cosmic ray detection in units of sigma is set by default to the relatively high value of **removecrhsingle-sigmalim = 20.0**. This is to avoid flagging the edges of the brightest sky lines as cosmic ray events. If cosmic ray hits are still present in the output spectrum, then reduce the value of this parameter while checking that the edges of sky lines are not flagged. In very few cases, you may need to set this threshold even higher.
- To control the sampling of the output 1D and 2D spectra, set the parameters **rectify-bin-lambda** and **rectify-bin-slit** appropriately.
- For UVB/VIS, sometimes improved sky subtraction may be obtained by increasing **sky-median-hsize** (for the MEDIAN method).
- The window used to define the local spatial profile is set via **stdextract-interp-hsize**. Increase this parameter if you find "gaps" in the 1D spectrum at the positions of bad pixels in order to interpolate over them.
- The output 1D spectrum has a third extension that stores quality flags. Please be aware that outliers in the flux extension may well be bad or interpolated values that are appropriately flagged in the third extension.
- Due to the improvement of the response determination achieved in this release we strongly discourage the use of response curves delivered by Calselector.

10.16 xsh_scired_slit_offset

This recipe reduces the science and sky frames observed in slit offset mode.

10.16.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_SLIT_OFFSET_ARM	1..N	any	100k
raw	SKY_SLIT_ARM	1..N	match	match
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	XSH_MOD_CFG_OPT_2D/AFC_ARM	1	1x1	400k
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	ORDER_TAB_EDGES/AFC_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM/DISP_TAB_AFC_ARM	?	1x1	400k
cdb	RESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	MRESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	ATMOS_EXT_ARM	?	-	-

In case of NIR data obtained with the JH filter the user must use an alternative spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

All OBJECT_SLIT_OFFSET_ARM and SKY_SLIT_ARM observations must have the same exposure time.

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_SLIT_OFFSET_ARM	1..N	any	100k
raw	SKY_SLIT_ARM	1..N	match	match
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	ORDER_TAB_EDGES/AFC_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM/DISP_TAB_AFC_ARM	1	1x1	400k
cdb	RESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	MRESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	ATMOS_EXT_ARM	?	-	-

10.16.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	<PREF>_ORDER2D_ARM	pro	Order by order 2D spectrum
1	<PREF>_ORDER1D_ARM	pro	Order by order 1D spectrum
2	<PREF>_MERGE2D_ARM	pro	Merged 2D spectrum
3	<PREF>_MERGE1D_ARM	pro	Merged 1D spectrum
4	<PREF>_SKY_ARM	pro	Sky frame
5	<PREF>_SKY_ORDER2D_ARM	pro	Order by order 2D sky spectrum
6	<PREF>_SKY_MERGE2D_ARM	pro	Merged 2D sky spectrum
7	<PREF>_FLUX_ORDER2D_ARM	pro	Order by order flux calibrated 2D spectrum
8	<PREF>_FLUX_ORDER1D_ARM	pro	Order by order flux calibrated 1D spectrum
9	<PREF>_FLUX_MERGE2D_ARM	pro	Merged flux calibrated 2D spectrum
10	<PREF>_FLUX_MERGE1D_ARM	pro	Merged flux calibrated 1D spectrum
11	<PREF>_WAVE_MAP_ARM	pro	Wave map frame
12	<PREF>_SLIT_MAP_ARM	pro	Slit map frame

The flux calibrated spectrum products are only written out if a response curve is supplied to the recipe.

10.16.3 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	1000	
rectify-kernel	tanh			

rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
stdextract-interp-hsize	30	0	1000	pix
combinenod-method	MEAN			
gen-sky	TRUE			

Note that the parameter **extract-method** should only be set to **LOCALIZATION** for this recipe. Other values will cause a recipe crash.

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.16.4 Recommendations and issues

- The input science and sky frames must all have the same exposure time. There must be at least the same number of sky observations as the number of science object observations, and the first N sky observations will be paired up with the N science observations.
- It is assumed that all the science frames have been observed such that the object is at the same position on the slit in all frames. If this is not the case, then the recipe reduction cascade is not appropriate, and the science frames should be reduced by this recipe individually.
- It is assumed that the sky does not vary significantly within each pair of object/sky observations. If this is not the case, then the recipe algorithms are not appropriate, and the results will suffer from systematic errors.
- The threshold for cosmic ray detection in units of sigma is set by default to the relatively high value of **removecrhsingle-sigmalim = 20.0**. This is to avoid flagging the edges of the brightest sky lines as cosmic ray events. If cosmic ray hits are still present in the output spectrum, then reduce the value of this parameter while checking that the edges of sky lines are not flagged. In very few cases, you may need to set this threshold even higher.
- To control the sampling of the output 1D and 2D spectra, set the parameters **rectify-bin-lambda** and **rectify-bin-slit** appropriately.
- The window used to define the local spatial profile is set via **stdextract-interp-hsize**. Increase this parameter if you find “gaps” in the 1D spectrum at the positions of bad pixels in order to interpolate over them.

- The output 1D spectrum has a third extension that stores quality flags. Please be aware that outliers in the flux extension may well be bad or interpolated values that are appropriately flagged in the third extension.
- Due to the improvement of the response determination achieved in this release we strongly discourage the use of response curves delivered by Calselector.

10.17 xsh_scired_slit_nod

This recipe reduces the science frames observed in slit nodding mode.

10.17.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	any	any
raw	OBJECT_SLIT_NOD_ARM	1..N	any	any
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	XSH_MOD_CFG_OPT_2D/AFC_ARM	1	1x1	400k
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	ORDER_TAB_EDGES/AFC_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM/DISP_TAB_AFC_ARM	?	1x1	400k
cdb	RESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	MRESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	ATMOS_EXT_ARM	?	-	-

In case of NIR data obtained with the JH filter the user must use an alternative spectral format table provided in the kit release tagged as SPECTRAL_FORMAT_TAB_JH_NIR.

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_SLIT_NOD_ARM	1..N	any	any
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	MASTER_FLAT_SLIT_ARM	1	match	match
cdb	ORDER_TAB_EDGES/AFC_SLIT_ARM	1	match	match
cdb	DISP_TAB_ARM/DISP_TAB_AFC_ARM	1	1x1	400k
cdb	RESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	MRESPONSE_MERGE1D_SLIT_ARM	?	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	ATMOS_EXT_ARM	?	-	-

10.17.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	<PREF>_ORDER2D_ARM	pro	Extracted 2D orders
1	<PREF>_ORDER1D_ARM	pro	Extracted 1D orders
2	<PREF>_MERGE2D_ARM	pro	Merged 2D spectrum
3	<PREF>_MERGE1D_ARM	pro	Merged 1D spectrum
4	<PREF>_WAVE_MAP_ARM	pro	Wave map frame
5	<PREF>_SLIT_MAP_ARM	pro	Slit map frame
6	<PREF>_FLUX_ORDER2D_ARM	pro	Flux calibrated order by order 2D frame
7	<PREF>_FLUX_ORDER1D_ARM	pro	Flux calibrated order by order 1D frame
8	<PREF>_FLUX_MERGE2D_ARM	pro	Flux calibrated 2D frame
9	<PREF>_FLUX_MERGE1D_ARM	pro	Flux calibrated 1D frame

10.17.3 Parameters

alias	default	min	max	units
pre-overscan-corr	1	0	6	
stack-method	median			
klow	5.0	0	100	
khigh	5.0	0	100	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	100	
rectify-kernel	tanh			
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
rectify-fast	TRUE			
localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
localize-nod-throw	0.0			
extract-method	NOD			
stdextract-interp-hsize	30	0	1000	pix
combinenod-throwlist	throwlist.asc			
combinenod-method	MEAN			
max-slit	5.7			arcsec
min-slit	-5.3			arcsec
correct-sky-by-median	TRUE			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR

rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.17.4 Recommendations and issues

- The input science frames must all have the same exposure time. Furthermore, there must be an even number of input science frames.
- It is assumed that the nod pattern is of the form AA..A BB..B CC..C DD..D etc. The individual sequences AA..A, BB..B, etc. can each have any number of exposures. These sequences are combined (**stack-method**, **klow**, **khigh**) to create the sequence A B C D etc. Then the combined images are subtracted in pairs to create [(A-B) - shifted(B-A)], [(C-D - shifted(D-C)], etc. Finally, all the combined and subtracted 2D spectra are combined (**combinenod-method**) into a single subtracted 2D spectrum.
- It is assumed that the sky does not vary significantly within each sequence AA..A, BB...B, etc. of science observations. If this is not the case, then the recipe algorithms are not appropriate, and the results will suffer from systematic errors.
- It is also assumed that the sky does not vary significantly between the pairs of images in the combined sequence A,B and C,D etc. However, to attempt to correct for this when it occurs, the recipe subtracts the median pixel value at each wavelength from the rectified A-B image before combination into the [(A-B) - shifted(B-A)] image. This correction can be switched off by setting **correct-sky-by-median = FALSE**.
- The threshold for cosmic ray detection in units of sigma is set by default to the relatively high value of **removecrhsingle-sigmalim = 20.0**. This is to avoid flagging the edges of the brightest sky lines as cosmic ray events. If cosmic ray hits are still present in the output spectrum, then reduce the value of this parameter while checking that the edges of sky lines are not flagged. In very few cases, you may need to set this threshold even higher.
- To control the sampling of the output 1D and 2D spectra, set the parameters **rectify-bin-lambda** and **rectify-bin-slit** appropriately.
- The window used to define the local spatial profile is set via **stdextract-interp-hsize**. Increase this parameter if you find "gaps" in the 1D spectrum at the positions of bad pixels in order to interpolate over them.
- The output 1D spectrum has a third extension that stores quality flags. Please be aware that outliers in the flux extension may well be bad or interpolated values that are appropriately flagged in the third extension.
- The **extract-method** default value is set to NOD for robustness reason. In certain cases **extract-method** set to LOCALIZATION may give better results.
- Due to the improvement of the response determination achieved in this release we strongly discourage the use of response curves delivered by Calselector.

10.18 xsh_scired_ifu_stare

This recipe reduce science object frames observed in IFU stare mode.

10.18.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_IFU_STARE_ARM	1..n	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	1	match	match
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	XSH_MOD_CFG_OPT_AFC_ARM	1	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	SKY_SUB_BKPTS_ARM	?	-	-
ref	IFU_CFG_TAB_ARM	?	-	-

Poly mode:

type	TAG	n	bin	RO
raw	OBJECT_IFU_STARE_ARM	1..n	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	1	match	match
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	WAVE_TAB_AFC_ARM	1	match	match
cdb	XSH_MOD_CFG_OPT_AFC_ARM	?+	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	SKY_SUB_BKPTS_ARM	?	-	-
ref	IFU_CFG_TAB_ARM	?	-	-

10.18.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	PREF_MERGE3D_DATA_OBJ_ARM	pro	Object merged 3D cube
1*	PREF_MERGE3D_TRACE_OBJ_ARM	pro	Object traces on merged cube
2	PREF_ORDER3D_DATA_OBJ_ARM	pro	Object Order-by-order 3d cube
3	IFU_CFG_COR_ARM	pro	ifu traces fit coefs differences
4*	IFU_MAP_SKY_AREA_ARM	pro	Image with obj,sky subtracted,flat fielded frame, wave map, ifu-slices map, ifu ID map (ID=1,2,3) RA map, DEC map, sky area map

10.18.3 Parameters

alias	default	min	max	units

stack-method	median			
klow	5	0	100	
khigh	5	0	100	
pre-overscan-corr	1	0	6	
crh-clip-kappa	5.0			
crh-clip-niter	5	0	20	
crh-clip-frac	0.7	0	100	
background-edges-margin	1	0	15	pix
background-poly-deg-x	9	0	15	
background-poly-deg-y	9	0	15	
background-poly-deg-kappa	10.0	0	100	
rectify-kernel	default			pix
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
compute-map	TRUE			
trace-obj	FALSE			
check-afc	TRUE			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.18.4 Recommendations and issues

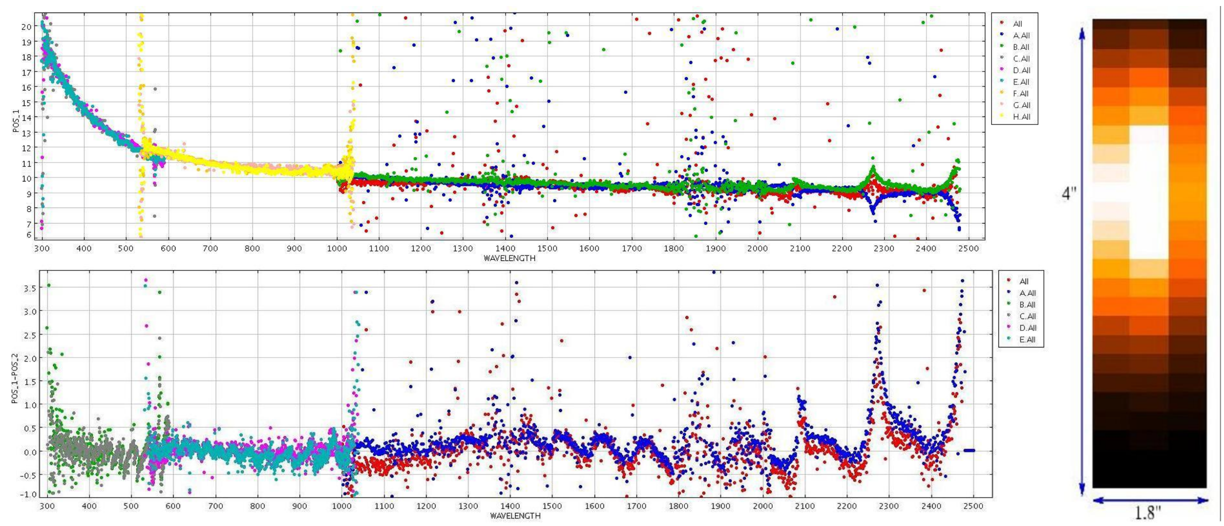


Figure 10.5: Upper left: combined IFU upper, central and lower traces for the three arms. The curvature is due to differential atmospheric dispersion as there is no ADC for the IFU. Lower left: residuals of lower-central and upper-central traces for a standard star used to verify accuracy. Right: a slice of the 3D cube at H_α. The spatial size of one pixel is 0.15”.

The user should verify that the object position traces on the three slices as function of wavelength do overlap within half a pixel. Alternatively the user may verify that the object position in the reconstructed cube is the same by moving along the wavelength direction. If this is not the case the user may try to reduce the Telluric standard observation usually taken in the same night when the object is observed and, using the optional input table tagged as IFU_CFG_TAB_ARM, modify the parameters specified by this in order to obtain proper object traces overlap, and then use the table with optimized coefficients as input of the science data reduction. This should allow to recover data reduction accuracy.

10.19 xsh_scired_ifu_offset

10.19.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_IFU_STARE_ARM	1,3..n	any	100k
raw	SKY_IFU_ARM	1,3..n	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	XSH_MOD_CFG_OPT_AFC_ARM	?+	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	IFU_CFG_TAB_ARM	?	-	-

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_IFU_OFFSET_ARM	1..n	any	100k
raw	SKY_IFU_ARM	1..n	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	WAVE_TAB_AFC_ARM	1	1x1	400k
cdb	DISP_TAB_AFC_ARM	1	1x1	400k
cdb	XSH_MOD_CFG_OPT_AFC_ARM	1	-	-
cdb	BP_MAP_NL_ARM	?	match	-
ref	BP_MAP_RP_ARM	?	match	-
ref	IFU_CFG_TAB_ARM	?	-	-

10.19.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	PREF_MERGE3D_DATA_OBJ_ARM	pro	Object merged 3D cube

1*	PREF_MERGE3D_TRACE_OBJ_ARM	pro	Object traces on merged cube
2	PREF_ORDER3D_DATA_OBJ_ARM	pro	Object Order-by-order 3d cube
3	IFU_CFG_COR_ARM	pro	ifu traces fit coefs differences
4	PREF_MERGE3D_DATA_SKY_ARM	pro	Sky merged 3D cube
5*	IFU_MAP_SKY_AREA_ARM	pro	Image with obj,sky subtracted,flat fielded frame, wave map, ifu-slices map, ifu ID map (ID=1,2,3) RA map, DEC map, sky area map

10.19.3 Parameters

alias	default	min	max	units
stack-method	median			
klow	5	0	100	
khigh	5	0	100	
crh-clip-kappa	0.3	0	20	
crh-clip-niter	5	0	100	
crh-clip-frac	5.0	0	1	
background-edges-margin	1	0	15	pix
background-poly-deg-x	9	0	15	
background-poly-deg-y	9	0	15	
background-poly-deg-kappa	10.0	0	100	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	1000	
rectify-kernel	default			
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	1	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
optimal-extract-kappa	-1.0			
compute-map	FALSE			
trace-obj	FALSE			
check-afc	TRUE			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.19.4 Recommendations and issues

10.20 xsh_cfg_recover

So far the X-shooter spectral format has been stable. In this case the user may adopt the reference solutions from the physical model configuration files distributed with the pipeline (xs_*_def.fits), with no need to re-optimize them. Otherwise, in case of significant spectral format shifts, the user can run this recipe to obtain an optimal instrument model configuration.

10.20.1 Input

UVB,VIS				
type	TAG	n	bin	RO
raw	FMTCHK_ARM	1	1x1	400k
ref	ARC_LINE_LIST_ARM	1	-	-
ref	XSH_MOD_CFG_TAB_ARM	1	-	-
usr	XSH_MEASCOORD_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match

The frame tagged as XSH_MEASCOORD_ARM is a frame containing the X and Y positions of a small set (12 is enough) of lines measured (with pixel accuracy) by hand by the user, by comparing their positions on a reference format-check frame where the frame tagged as XSH_MOD_CFG_TAB_ARM was a good model configuration file (more details on this procedure are in [14,15]).

Notes: The XSH_MOD_CFG_ARM and XSH_MEASCOORD_ARM are the one indicated as appropriate for the data reduction (may not have the same file name as the one here provided).

NIR			
type	TAG	n	bin
raw	FMTCHK_NIR_ON	1	1x1
raw	FMTCHK_NIR_OFF	1	1x1
ref	ARC_LINE_LIST_NIR	1	-
ref	XSH_MOD_CFG_TAB_NIR	1	-
usr	XSH_MEASCOORD_NIR	1	-

The frame tagged as XSH_MEASCOORD_NIR is a frame containing the X and Y positions of a small set (12 is enough) of lines measured (with pixel accuracy) by hand by the user, by comparing their positions on a reference format-check frame where the frame tagged as XSH_MOD_CFG_NIR was a good model configuration file (more details on this procedure are in [14,15]).

Notes: The XSH_MOD_CFG_NIR and XSH_MEASCOORD_NIR are the one indicated as appropriate for the data reduction (may not have the same file name as the one here provided).

The recipe man page clarify the recipe input/output:

```
Input Frames :
   - [UVB, VIS] A RAW frame (Format = RAW, Tag = FMTCHK_arm)
   - [NIR] Two RAW frames (Format = RAW, Tag = FMTCHK_arm_ON,FMTCHK_arm_OFF)
   - [UVB,OPTIONAL-required if trace-orders=TRUE] a RAW frame (Format = RAW, Tag = ORDERDEF_D2_UVB)
   - [VIS,OPTIONAL-required if trace-orders=TRUE] a RAW frame (Format = RAW, Tag = ORDERDEF_arm)
   - [NIR,OPTIONAL-required if trace-orders=TRUE]
     Two RAW frames (Format = RAW, Tag = ORDERDEF_arm_ON, ORDERDEF_arm_OFF)
   - [UVB,VIS,OPTIONAL] A master bias (Format = PRE, Tag = MASTER_BIAS_arm)
   - [UVB,VIS,OPTIONAL] A master dark (Format = PRE, Tag = MASTER_DARK_arm)
   - [OPTIONAL-Required if trace-orders=TRUE]
    A spectral format table (Format = TABLE, Tag = SPECTRAL_FORMAT_TAB_arm)
   - [OPTIONAL-Required if trace-orders=FALSE] an order table (Format = TABLE, Tag = ORDER_TAB_CENTR_arm)
   - A ref. line list (Format = TABLE, Tag = ARC_LINE_LIST_arm)
   - A model cfg. file (Format = TABLE, Tag = XSH_MEASCOORD_arm)
 Products :
   - if first-anneal=FALSE & trace-orders=FALSE & last-step=FALSE nothing
   - if first-anneal=TRUE & last-step=FALSE an optimized model configuration, PRO.CATG=XSH_MOD_CFG_TAB_arm
   - if first-anneal=TRUE & trace-orders=TRUE, last-step=TRUE an order table, PRO.CATG=ORDER_TAB_CENTR_arm
   - if last-step=TRUE an optimized model configuration, PRO.CATG=CONF_OPT_arm
     a quality control table, PRO.CATG=MODEL_GUESS_XY_arm
     the model theoretical map corresponding to the optimized model config,
     PRO.CATG=THEO_TAB_MULT_arm, THEO_TAB_IFU_arm, THEO_TAB_SING_arm.
```

10.20.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	XSH_MOD_CFG_FAN_ARM	qc	model cfg after 1st annealing
1	XSH_MOD_CFG_OPT_ARM	cdb	model cfg optimized
2	MODEL_GUESS_XY_ARM	qc	model guess x,y line positions
3	THEO_TAB_SING_ARM	qc	model central pinhole positions for poly mode
4	THEO_TAB_MULT_ARM	qc	model pinhole positions for poly mode

The XSH_MOD_CFG_OPT_ARM product contains several raws each with a model configuration parameter. For each parameter the user can specify a proper values to set the corresponding Low_limit a High_Limit, Compute_Flag, Parameter_Name, Parameter_Units.

The MODEL_GUESS_XY_ARM table contains the following columns:

WAVELENGTH	wavelength of sampling line [nm]
XG	X model prediction [pix]
YG	Y model prediction [pix]
ABS_ORD	absolute order
XC	X position corrected by user setting and order position [pix]
YC	Y position corrected by user setting and order position [pix]

The THEO_TAB_MULT_ARM tables contain the following columns:

Wavelength	wavelength of sampling line [nm]
Order	absolute order
slit_index	slit (pinhole) index
slit_position	slit (pinhole) position [arcsec]
detector_x	detector x predicted position [pix]
detector_y	detector y predicted position [pix]

Similar columns are in the THEO_TAB_SING_ARM table (but the slit value is 4, corresponding to the central pinhole).

Additional output:

```
FILE                                  PRO.CATG
MEASURE_LINE_POS_XY_ARM.fits          MEASURE_LINE_POS_XY_ARM
MODEL_GUESS_XY_ARM.fits
XSH_MOD_CFG_FAN_ARM.fits
model_THE1.fits
model_THE9.fits
new_xs_vis_def_com1.3.fits
short_spec_form.fits                  SPECTRAL_FORMAT_TAB_ARM
spec_form.fits                        SPECTRAL_FORMAT_TAB_ARM
```

ima_raw.fits is the format-check image in pre format, where pixel of intensity smaller than **ima_tresh_min** are set to 0.

The MODEL_GUESS_XY_ARM.fits table is the same as the MODEL_GUESS_XY_ARM product.

MEASURE_LINE_POS_XY_ARM.fits table contains the following columns:

Wavelength	wavelength of sampling line [nm]
XG	X model prediction [pix]
YG	Y model prediction [pix]
Order	absolute order
Xthpre	X model prediction [pix]
Ythpre	Y model prediction [pix]
Slit_position	slit (pinhole) position [arcsec]
XDiffGauss	extra column to fit to function interface [pix]
YDiffGauss	extra column to fit to function interface [pix]
XDiffPoly	extra column to fit to function interface [pix]
YDiffPoly	extra column to fit to function interface [pix]
SigmaXGauss	extra column to fit to function interface [pix]
SigmaYGauss	extra column to fit to function interface [pix]
Slit_index	slit (pinhole) index
XM	X model prediction after annealing [pix]
YM	Y model prediction after annealing [pix]
RESIDX	Residual corresponding to XM [pix]
RESIDY	Residual corresponding to YM [pix]

The model_THE9.fits table is the same as the THEO_TAB_MULT_ARM product. The model_THE1.fits table is the same as the THEO_TAB_SING_ARM product. The new_xs_vis_def_jun08.fits table is the same as the CONF_OPT_ARM product.

10.20.3 Quality control

The user may compare positions of positions in the different tables with the ones of the lines in the image in pre format (ima_raw.fits).

10.20.4 Parameters

alias	default	min	max

first-anneal	FALSE		
arm	vis		
name_i	line_xy_ord.txt		
niter	100000		
coord_frame	1		
last-step	FALSE		
plot	FALSE		
ima_tresh_min	40.0		
cor_prescan	FALSE		
method	safefit		
offx	0.0		
offy	0.0		
slit	5		
gfit_box_sy	20		
peak_line_fwhm	4		
peak_kappa	5.0		
peak_factor	10.0		
peak_match_x	10		
peak_match_y	20		
pm_ord_sel	1		
pm_radius	20.0		
pm_tolerance	0.1		
anneal_niter	1000		

The most important parameters are **first-anneal**, **niter**, **method**, **anneal_niter**. **method** should be set to 'safefit'. **niter** and **anneal_niter** should be large enough (10^4) to get good accuracy, but not too much (10^7) not have reasonable computation times.

10.20.5 Recommendations and issues

Input frames for the xsh_cfg_recover, xsh_predict, xsh_orderpos, xsh_2dmap recipes are always unbinned data. This means that the results refer to the unbinned spectral format. The other recipes accept data of any binning. If a pipeline product table was generated processing an unbinned frame and is required to reduce binned data, the corresponding pipeline recipe automatically applies proper correction for the bin setting. In this case, the input calibration instrument setting does not need to match with that of the corresponding recipe input frame (except obviously for the value of the SEQ.ARM FITS header keyword).

reduction of IFU data with atmospheric dispersion correction

10.21 xsh_geom_ifu

The recipe is aimed at tracing the position of a point-like object in the IFU and derive shifts along the slitlets as a function of wavelength with respect to a reference position. These shifts will correct any distorsion due the atmospheric dispersion, the wavelength solution and the optics of the IFU. This recipe generates three tables (PRO.CATG=OFFSET_TAB_UP(CEN, DOWN)_ARM) containing the relative shift between the different slices of the IFU. It requires in input an observation of a spectrophotometric/telluric point-like standard star. The recipe can produce a cube with the input star using the produced tables. As one step of the data reduction involves the order rectification, which is performed applying previously computed polynomial transformations, that are different for each arm, also the table generated is arm specific.

10.21.1 Input

Physical model mode (recommended):

UVB,VIS				
type	TAG	n	bin	RO
raw	TRACE_ARM	1	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	1	match	match
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	XSH_MOD_CFG_OPT_AFC_ARM	1	1x1	400k

NIR			
type	TAG	n	bin
raw	TRACE_NIR	1	any
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
cdb	MASTER_FLAT_IFU_NIR	1	match
cdb	ORDER_TAB_AFC_IFU_NIR	1	match
cdb	XSH_MOD_CFG_OPT_AFC_NIR	1	1x1

Poly mode:

UVB,VIS				
type	TAG	n	bin	RO
raw	TRACE_ARM	1	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	1	match	match
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	WAVE_TAB_2D_ARM	1	1x1	400k
cdb	DISP_TAB_AFC_ARM	1	1x1	400k

NIR			
type	TAG	n	bin
raw	TRACE_NIR	1	any
ref	SPECTRAL_FORMAT_TAB_NIR	1	-
cdb	MASTER_FLAT_IFU_NIR	1	match
cdb	ORDER_TAB_AFC_IFU_NIR	1	match
cdb	WAVE_TAB_2D_NIR	1	1x1
cdb	DISP_TAB_AFC_NIR	1	1x1

10.21.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	OFFSET_TAB_DOWN_IFU_ARM	cdb	Table with slice relative positions
1	OFFSET_TAB_CEN_IFU_ARM	cdb	Table with slice relative positions

2	OFFSET_TAB_UP_IFU_ARM	cdb	Table with slice relative positions
3*	TELL_IFU_MERGE3D_IFU_ARM	cdb	Reconstructed cube
4*	TELL_IFU_GEOM_IFU_FF_ARM_TRACE_OBJ_ARM	cdb	Table with object traces for quality control

10.21.3 Quality control

10.21.4 Parameters

alias	default	min	max
pre-overscan-corr	1	0	6
stack-method	median		
crh-clip-kappa	5.0	0	6
crh-clip-niter	5	0	20
crh-clip-frac	0.7	0	1
background-edges-margin	1	0	15
background-poly-deg-x	9	0	15
background-poly-deg-y	9	0	15
background-poly-deg-kappa	10.0	0	100
removecrhsingle-sigmalim	20.0	0	200
removecrhsingle-flim	2.0	0	20
removecrhsingle-niter	4	0	1000
rectify-kernel	default		
rectify-radius	2.0	2	100
rectify-bin-lambda	-1.0	0	210
rectify-bin-slit	-1.0	0	6
localizeifu-method	MANUAL		
localizeifu-chunk-nb	10	1	1000
localizeifu-thresh	0.1	0	1
localizeifu-deg-lambda	0	0	10
localizeifu-slit-position	0.0	-7	7
localizeifu-slit-hheight	2.0	0	7
localizeifu-kappa	3.0	0	20
localizeifu-niter	3	0	100
localizeifu-use-skymask	FALSE		
correctifu-niter	-1.0		
correctifu-lambdaref	-1.0		
correctifu-lambdaref-hsize	2.5		
do-cube	TRUE		
compute-map	TRUE		
check-afc	TRUE		
flat-method	blaze		

10.21.5 Recommendations and issues

In order to check the quality of the reduction the user should set the parameter **do-cube** to TRUE and check the alignment of the 'self-aligned' cube TELL_CUBE_MERGE3D_IFU_ARM plotting the traces of the from the file TELL_IFU_GEOM_IFU_FF_ARM_TRACE_ARM. See Figure 10.6 for an example of a succesful alignment. The quality of the result is strongly dependent on the accuracy of the localization of the traces in the three slitlets. If the traces are not well aligned, the user may increase the parameter **–localize-ifu-chunk-hsize** in order to obtain a higher signal to noise in localization or mask the slit edges using **–localizeifu-slitlow(up)-edges-mask**. The user may also increase (or possibly decrease) the number of iterations with the parameter **correctifu-niter**.

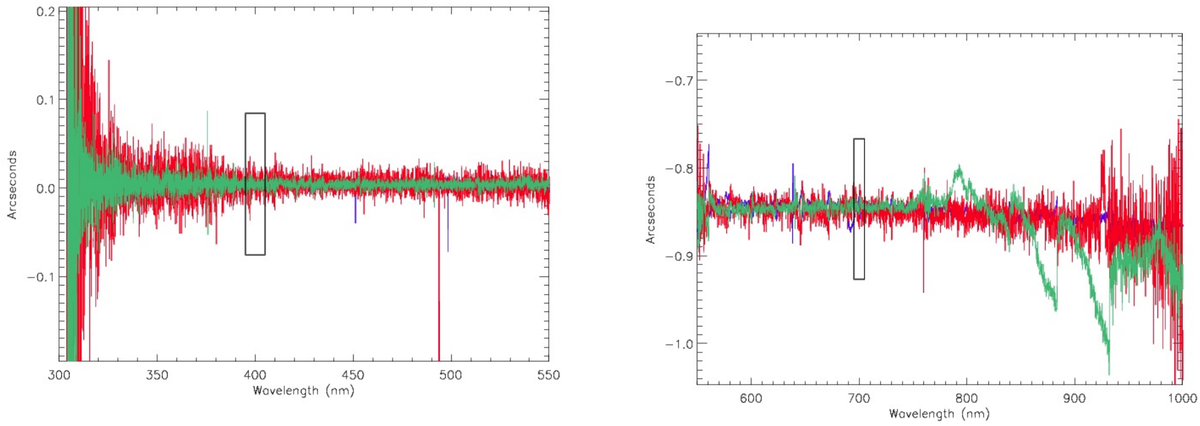


Figure 10.6: *Left* Well aligned traces of the three slitlets of a self calibrated cube (UVB arm) taken at low airmass (1.05). *Right* Badly aligned races of the three slitlets of a self calibrated cube (VIS arm) taken at high airmass (1.53). These traces may be better aligned increasing the parameter **correctifu-niter**. In both plots the rectangular box has the dimension of a pixel for a default reduction (0.16 $''$).

10.22 xsh_scired_ifu_stare_drl

This recipe reduce science object frames observed in IFU stare mode and corrects for the effects of atmospheric dispersion straightening the traces in the 2D wavelength calibrated frame.. The rectified frames are corrected using the OFFSET tables (PRO.CATG=SLICE_OFFSET_TABLE_VIS) produced by the recipe xsh_geom_ifu. This recipe should be executed only after having executed xsh_geom_ifu with a suitable telluric standard star. Due to the airmass difference between the telluric star and the science target, this correction cannot be perfect, therefore a residual curvature in the 2D frame will still be present.

10.22.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	`OBJECT_IFU_STARE_ARM`	1..n	any	100k
ref	`SPECTRAL_FORMAT_TAB_ARM`	1	-	-
cdb	`MASTER_BIAS_ARM`	1	match	match
cdb	`MASTER_FLAT_IFU_ARM`	1	match	match
cdb	`ORDER_TAB_AFC_IFU_ARM`	1	match	match
cdb	`XSH_MOD_CFG_OPT_AFC_ARM`	1	-	-
cdb	`OFFSET_TAB_UP_IFU_ARM`	1	-	-
cdb	`OFFSET_TAB_CEN_IFU_ARM`	1	-	-
cdb	`OFFSET_TAB_DOWN_IFU_ARM`	1	-	-
*cdb	`BP_MAP_NL_ARM`	?	match	-
*ref	`BP_MAP_RP_ARM`	?	match	-

ref	SKY_SUB_BKPTS_ARM	?	-	-

Poly mode:

type	TAG	n	bin	RO
raw	OBJECT_IFU_STARE_ARM	1..n	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	1	match	match
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	WAVE_TAB_AFC_ARM	1	match	match
cdb	DISP_TAB_AFC_ARM	1	-	-
cdb	OFFSET_TAB_UP_IFU_ARM	1	-	-
cdb	OFFSET_TAB_CEN_IFU_ARM	1	-	-
cdb	OFFSET_TAB_DOWN_IFU_ARM	1	-	-
*cdb	BP_MAP_NL_ARM	?	match	-
*ref	BP_MAP_RP_ARM	?	match	-

10.22.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	TELL_IFU_ORDER2D_DOWN_IFU_ARM	pro	Order by order extracted 'down' IFU slice frame
1	TELL_IFU_MERGE2D_DOWN_IFU_ARM	pro	Merged 'down' IFU slice frame
2	TELL_IFU_ORDER2D_CEN_IFU_ARM	pro	Order by order extracted 'cen' IFU slice frame
3	TELL_IFU_MERGE2D_CEN_IFU_ARM	pro	Merged 'cen' IFU slice frame
4	TELL_IFU_ORDER2D_UP_IFU_ARM	pro	Order by order extracted 'up' IFU slice frame
5	TELL_IFU_MERGE2D_UP_IFU_ARM	pro	Merged 'up' IFU slice frame
6	TELL_IFU_MERGE3D_IFU_ARM	pro	Reconstructed object cube
7	TELL_IFUON_ARM	pro	Bias (Off) corrected IFU frame
8	TELL_IFU_SUB_BACK_IFU_ARM	pro	Inter-order background corrected obj IFU frame
9	TELL_IFU_DIV_FF_ARM	pro	Flat fielded obj IFU frame
10	TELL_IFU__TRACE_OBJ_ARM	pro	Table with object traces for quality control

10.22.3 Parameters

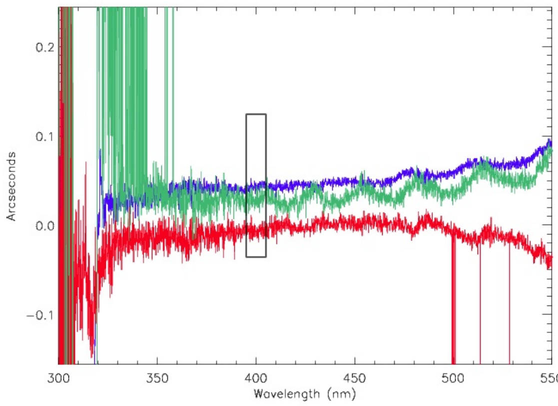
alias	default	min	max	units
pre-overscan-corr	1	0	6	
crh-clip-kappa	5.0			
crh-clip-niter	5	0	20	
crh-clip-frac	0.7	0	100	
background-edges-margin	1	0	15	pix
background-poly-deg-x	9	0	15	
background-poly-deg-y	9	0	15	
background-poly-deg-kappa	10.0	0	100	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-niter	4	0	20	
rectify-kernel	default			pix
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	0	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
compute-map	TRUE			
shift-offsettab-low	0.0	0.0	0.15	
shift-offsettab-up	0.0	0.0	0.15	
check-afc	TRUE			
flat-method	blaze			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.22.4 Recommendations and issues

Read the list of known problems for this recipe in Section 6. If the traces in the data cube have a residual misalignment of less than a pixel, this offset can be corrected with the parameter **shift-offsettab-low(up)**. (see Figure 10.7). Sometimes the traces of the lateral slitlets are very noisy, especially at short wavelengths in UVB (see Figure 10.7). This may be due to the lower S/N of the source in this range or to the fact that the source is near to the order edge which is very noisy in this region. The automatic localization is then heavily influenced by this noise.



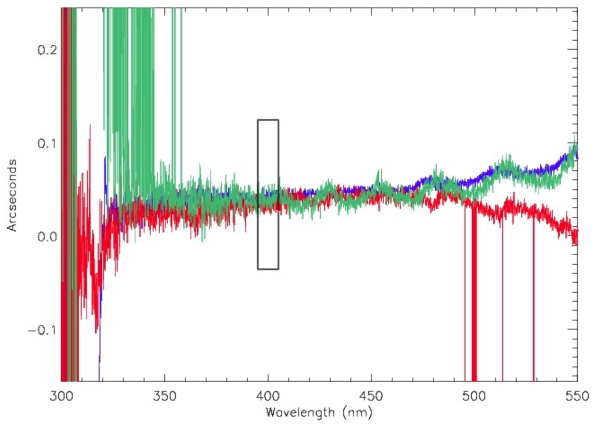


Figure 10.7: *Left* (UVB arm) Traces of a data cube obtained with xsh_scired_ifu_stare_drl. Note that a residual offset is present between the traces *Right* The same cube after applying appropriate offsets with the parameter **shift-offsettab-low(up)**during the reduction. In both plots the rectangular box has the dimension of a pixel for a default reduction (0.16 ″).

10.23 xsh_scired_ifu_offset_drl

This recipe reduce science object frames observed in IFU offset mode. The rectified frames are corrected using the OFFSET tables (PRO.CATG=SLICE_OFFSET_TABLE_VIS) produced by the recipe xsh_geom_ifu. This recipe should be executed only after having executed xsh_geom_ifu with a suitable telluric standard star.

10.23.1 Input

Physical model mode (recommended):

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_IFU_OFFSET_ARM	1..n	any	100k
raw	SKY_IFU_ARM	1..n	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	OFFSET_TAB_DOWN_IFU_ARM	1	match	match
cdb	OFFSET_TAB_CEN_IFU_ARM	1	match	match
cdb	OFFSET_TAB_UP_IFU_ARM	1	match	match
cdb	DISP_TAB_AFC_ARM	1	1x1	400k
cdb	XSH_MOD_CFG_OPT_AFC_ARM	1	-	-
*cdb	BP_MAP_NL_ARM	?	match	-
*ref	BP_MAP_RP_ARM	?	match	-

Poly mode:

UVB,VIS,NIR				
type	TAG	n	bin	RO
raw	OBJECT_IFU_OFFSET_ARM	1..n	any	100k
raw	SKY_IFU_ARM	1..n	any	100k
ref	SPECTRAL_FORMAT_TAB_ARM	1	-	-
cdb	MASTER_BIAS_ARM	?	match	match
cdb	MASTER_FLAT_IFU_ARM	1	match	match
cdb	ORDER_TAB_AFC_IFU_ARM	1	match	match
cdb	OFFSET_TAB_DOWN_IFU_ARM	1	match	match
cdb	OFFSET_TAB_CEN_IFU_ARM	1	match	match
cdb	OFFSET_TAB_UP_IFU_ARM	1	match	match
cdb	WAVE_TAB_AFC_ARM	1	1x1	400k
cdb	DISP_TAB_AFC_ARM	1	1x1	400k
*cdb	BP_MAP_NL_ARM	?	match	-
*ref	BP_MAP_RP_ARM	?	match	-

10.23.2 Output

UVB,VIS,NIR			
ID	PRO.CATG	type	Note
0	PREF_MERGE3D_IFU_ARM	pro	Object merged 3D cube
1*	PREF_MERGE3D_TRACE_OBJ_ARM	pro	Object traces on merged cube

10.23.3 Parameters

alias	default	min	max	units
stack-method	median			
klow	5	0	100	
khigh	5	0	100	
crh-clip-kappa	0.3	0	20	
crh-clip-niter	5	0	100	
crh-clip-frac	5.0	0	1	
background-edges-margin	1	0	15	pix
background-poly-deg-x	9	0	15	
background-poly-deg-y	9	0	15	
background-poly-deg-kappa	10.0	0	100	
removecrhsingle-sigmalim	20.0	0	200	
removecrhsingle-flim	2.0	0	20	
removecrhsingle-niter	4	0	1000	
rectify-kernel	tanh			pix
rectify-radius	2.0	2	100	pix
rectify-bin-lambda	-1.0	1	210	nm
rectify-bin-slit	-1.0	0	6	arcsec
localize-method	MANUAL			
localize-chunk-nb	10	1	1000	
localize-thresh	0.1	0	1	
localize-deg-lambda	0	0	10	
localize-slit-position	0.0	-7	7	arcsec
localize-slit-hheight	2.0	0	7	arcsec
localize-kappa	3.0	0	20	
localize-niter	3	0	100	
localize-use-skymask	FALSE			
optimal-extract-kappa	-1.0			
compute-map	FALSE			
trace-obj	FALSE			
check-afc	TRUE			

If the user does not set different values the recipe automatically sets the following arm dependent values for the corresponding parameters:

parameter	default	actual used value		
		UVB	VIS	NIR
rectify-bin-lambda	-1.0	0.02	0.02	0.06
rectify-bin-slit	-1.0	0.16	0.16	0.21

10.23.4 Recommendations and issues

11 Algorithms and recipe details

In this Section we describe the main algorithms implemented in the X-Shooter pipeline recipes. Relevant data reduction parameters are typed in **bold** face.

11.1 Frame preparation (conversion to PRE format)

The raw input images are rotated so as to have a common orientation for each arm (NIR data are rotated by 90° clockwise, UVB data are rotated by 180°, VIS data orientation is not changed). For UVB/VIS only, if the parameter **pre-overscan-corr** is greater than zero, then the overscan region(s) are used to determine the bias level, which is subtracted from the image values, otherwise the bias level is subtracted using the master bias frame. The on-frame NIR images are corrected by the off-frame images, which includes the subtraction of the unknown bias level and dark level. For each image, a map of pixel uncertainties is calculated (see Sec. 11.2) and stored in a second extension. A third extension is also created to store bad pixel identification codes, and this is when saturated or negative pixels are flagged (including pixels with extrapolated flux for NIR), and also when bad pixels from a reference bad pixel map are included.

11.2 Detector noise model

Error extensions are created for each frame during frame preparation. The uncertainty σ_i (ADU) associated to a pixel i is calculated via:

$$\sigma_i = \sqrt{\sigma_0^2 + \frac{D_i}{G}}$$

where D_i is the raw image pixel value (bias level corrected in the case of UVB/VIS frames; for NIR frames D_i includes the unkown bias level), σ_0 is the detector read out noise (ADU), and G is the gain (e$^-$/ADU). In the case of UVB/VIS, the readout noise and gain are taken from the header keywords DET OUT1 RON and DET OUT1 CONAD, respectively. In the case of NIR, the read out noise is taken from a look-up table depending on the exposure integration time, and the gain is fixed to 2.12 e$^-$/ADU. For bias frames, we assume $\sigma_i = \sigma_0$ for all pixels. The pixel uncertainties at all stages of the pipeline are propagated using the standard error propagation formulae. Note that the above noise model is also applied to the extrapolated pixel values in the NIR images, which leads to an under-estimate of the corresponding pixel uncertainties.

11.3 Bad pixel code conventions

The bad pixel codes used in the bad pixel masks and 3rd extensions in the X-shooter pipeline are as follows:

Bit #	Flag Value	Quality condition
0	0	Pristine good pixel
1	1	Affected by telluric feature (corrected)
2	2	Affected by telluric feature (uncorrected)
3	4	Ghost/stray light at > 10% intensity level
4	8	Electronic pickup noise
5	16	Cosmic ray (removed)
6	32	Cosmic ray (unremoved)
7	64	Low QE pixel (< 20% of the average sensitivity (e.g. defective CCD coating, vignetting...)

8	128	Calibration file defect (if pixel is flagged in any calibration file)
9	256	Hot pixel ($> 5\ \sigma$ median dark)
10	512	Dark pixel (permanent CCD charge trap)
11	1024	Questionable pixel (lying above a charge trap which may have affected it)
12	2048	Detector potential well saturation (signal irrecoverable, but known to exceed the max. e^- number)
13	4096	A/D converter saturation (signal irrecoverable, but known to exceed the A/D full scale signal)
14	8192	Permanent camera defect (such as blocked columns, dead pixels)
15	16384	Bad pixel that does not fit into any other category
16	32768	Non-linear response pixel
17	65536	User defined - non-spatial uniformity
18	131072	User defined - divisor zero
19	262144	User defined - out of nod
20	524288	User defined - missing data
21	1048576	User defined - Extrapolated flux in NIR (for exposure times $\geq$1.2 s)
22	2097152	User defined - Raw pixel value is zero or negative
23	4194304	User defined - Interpolated flux during standard extraction
...	...	User defined
30	2^{29}	Outside data range (outside of spectral range, inactive detector area, mosaic gap, ...)

Table 11.1: Possible bad pixel codes.

11.4 Bias level determination from the overscan regions

The parameter **pre-overscan-corr** defines how the bias level is determined for each UVB/VIS image (pre and overscan regions are defined in the image headers). The acceptable values are:

value	help
0	No bias level correction using the overscan regions. The bias level is taken from the master bias instead.
1	The bias level is calculated as the 3-sigma clipped mean of the overscan region.
2	The bias level is calculated as the 3-sigma clipped mean of the prescan region.
3	The bias level is calculated as the mean of the two 3-sigma clipped mean values calculated in options 1 and 2.
4	The bias level is calculated on a row by row basis as the median of the corresponding row in the overscan region.
5	The bias level is calculated on a row by row basis as the median of the corresponding row in the prescan region.
6	The bias level is calculated on a row by row basis as the mean of the two median values calculated in options 4 and 5.

11.5 Kappa-sigma-clipped mean

This general algorithm computes a kappa-sigma-clipped mean and standard deviation through iterative kappa-sigma-clipping of outliers. On the first iteration, the algorithm uses all the non-flagged pixels (via the bad pixel map) to calculate an initial value for the sample *mean* and standard deviation *stdev*. Then the algorithm rejects all pixels with values outside the range *mean* $\pm$ **kappa** $\times$ *stdev*, and recalculates new values for the sample *mean* and standard deviation *stdev*. This process is iterated until **niter** iterations have been performed or until the *mean* does not change (by more than an absolute **tolerance**) between consecutive iterations.

11.6 Master Bias creation (xsh_mbias)

The raw frames are prepared (see 11.1). Then the mean bias level is calculated on each frame using a kappa-sigma-clipped mean, and this is subtracted from the respective frame. The bias-level corrected frames are combined using a median (**stack-method** = "median") or a kappa-sigma-clipped mean (**stack-method** = "mean"). Finally, the mean of the bias level from each bias frame is added back into the master bias frame.

Then the quality control parameters are computed: the bias level, the X and Y structures, the detector Read Out

Noise (see 11.8) and Fixed Pattern Noise (see 11.7), are computed in a user defined region.

11.7 Fixed Pattern Noise determination - random method (xsh_mbias)

The Fixed Pattern Noise (FPN) is determined using the first and second input raw bias frames. The standard deviation associated with the kappa-sigma-clipped mean is computed in two iterations for each frame. In the first iteration the frame median and standard deviation (stdev) are computed. Pixel values that lie more than 3*stdev from the median are rejected. Then an updated value for the standard deviation is computed. The read out noise of the frame, ron, is equal to this standard deviation, divided by $\sqrt{2}$. Then a robust standard deviation, $stdev_{robust}$ is computed on the difference of the first two bias frames shifted relative to each other by 10 rows and 10 columns. Again the median of this difference image is calculated and pixels values that lie more than $\sqrt{50}$ from the median value are rejected. Finally the mean of the non-rejected pixels on the difference image is calculated and divided by $\sqrt{2}$ to yeld a robust standard deviation. Finally the fixed pattern noise associated to the master bias frame is given by $\sqrt{stdev_{robust}^2 - ron^2}$.

11.8 Bias Read Out Noise determination

From the input set of bias frames the first two are extracted. Then the random noise and its error are computed by adopting **random_nsamples** sampling square boxes each of size **random_sizex** in two regions (**ref1_llx, ref1_lly, ref1_urx, ref1_ury** and **ref2_llx, ref2_lly, ref2_urx, ref2_ury**). The detector read out noise is equal to the random noise divided by $\sqrt{2}$.

11.9 Frame structure determination

The master bias frame is collapsed along the X or Y direction to determine its Y or X structure. The X (Y) structure of a region on the detector is determined as follows. On the master bias statistics are computed excluding the contribution of bad pixels. A sub-image is extracted according to the values of the parameters **ref1_llx,ref1_lly,ref1_urx,ref1_ury** set by the user. This region is then collapsed along the Y (X) direction. A clipped standard deviation is computed by rejecting values that lie outside $\pm$ 2 ADU from the mean evaluated on the full frame. As some X-shooter master bias settings show a structure the pipeline allows the user to compute the structure in two different regions.

11.10 Master Dark generation (xsh_mdark)

This recipe only accepts raw dark frames with the same exposure time. The raw frames are prepared (see 11.1). The mean level of each dark frame is calculated using a kappa-sigma-clipped mean, and this is subtracted from the respective frame. The frames are then combined using a median (**stack-method** = "median") or a kappa-sigma-clipped mean (**stack-method** = "mean"). Finally, the mean of the mean levels is added back into the master dark frame. The master dark frame is only normalised to an exposure time of 1s for UVB/VIS.

11.11 Noisy pixels detection in NIR dark frames

The xsh_mdark recipe will attempt to detect the noisy pixels in NIR dark frames. The recipe does this by computing the average, median and *standard deviation* for each pixel over the stack of frames. It then flags noisy pixels as those pixels whose values differ with from the median at each pixel over the stack of frames by more than **noise-clip-kappa** × *standard deviation*.

Additional noisy pixels are computed by building a cube of dark frames and examining the noise variations in each pixel. The pixel intensities that lie outside the range [**noise-lower-rejection**,**noise-higher-rejection**] are clipped and hence a *mean* and clean *standard deviation* are computed. If a pixel value deviates from a clean *mean* pixel value by more than **noise-clip-kappa** × *standard deviation*, the pixel is declared as bad.

The detected noisy pixels are flagged in the bad pixel mask with code 8. The total number of noisy pixels is logged and saved as the QC parameter (QC.BP-MAP.PICKUP.NOISEPIX).

11.12 Detection of arc lines (xsh_predict, xsh_2dmap, xsh_flexcomp)

At various stages in the pipeline, it is necessary to detect the emission lines that are present in an arc-line spectrum image. This is done using the following general algorithm.

Given a set of expected (or "guess") arc-line positions in an arc-lamp image (e.g. procedant from the physical instrument model), the algorithm performs a search for each arc-line in the image using a square window of half-size **detectarclines-search-win-hsize** pixels. Each square window is extracted from the image and filtered with a running box-car median of half-size **detectarclines-running-median-hsize** in order to remove cosmic ray hits from the image window. If the parameter **detectarclines-running-median-hsize** is set to zero, then this median filtering is not performed. An initial estimate of the detected line position in the image window is taken as the pixel with the maximum value in this filtered window. This new line position is used to extract a new square window of half-size **detectarclines-fit-win-hsize** from the original unfiltered image, and a refined line position is determined via a 2D Gaussian fit or a simple centroid method (with the method determined by the parameter **detectarclines-find-lines-center**).

Arc-lines are rejected from the list of detected lines if their coordinates could not be derived for some reason (i.e. the fit fails), or if the S/N ratio of the line is less than the threshold defined by the parameter **detectarclines-min-sn**, where the S/N ratio is defined as the counts of the central pixel of the line to the uncertainty of the pixel value. In the case of 9-pinhole data, arc-lines are rejected if less than 7 out of the 9 pinhole arc-lines have been detected.

11.13 Instrument physical model optimization (xsh_predict, xsh_cfg_recover)

The physical model parameters are optimised so that the mapping:

$$(\lambda, s, O) \rightarrow (x_{mod}, y_{mod})$$

results in the minimum possible value of:

$$\sum (x_{mod} - x_{meas})^2 + (y_{mod} - y_{meas})^2$$

where s is the entrance slit position, O the spectral order, x_{mod} and y_{mod} are the physical model predicted detector co-ordinates, x_{meas} and y_{meas} are the measured co-ordinates of the same wavelengths in calibration lamp exposures and the summation is over all matched calibration features. This is analogous to the fitting of a conventional polynomial to calibration lamp data, however the optimisation process here requires a more flexible minimisation algorithm, in this case Simulated Annealing. This is described in greater detail in [3]. The maximum number of iterations of the annealing process is controlled by **model-maxit** and the annealing factor is controlled by **model-anneal-factor**. The physical model scenario is controlled by **model-scenario** and corresponds to the specific recipe, and so its value should not be changed.

Note that no iterative sigma-clipping rejection is applied in the minimisation of the above sum in the case of the physical model (i.e. the values of the parameters **detectarclines-clip-sigma, detectarclines-clip-niter, detectarclines-clip-frac** are ignored).

11.14 Instrument polynomial model fitting (xsh_predict, xsh_2dmap, xsh_flexcomp)

The polynomial instrumental model, which is a wavelength solution expressing the X and Y pixel positions as a function of wavelength λ, order number n, and slit position s, is fitted to the set of detected lines in the arc-line image. The polynomial model employs Chebyshev polynomials, and the degree of the polynomial in each parameter λ, n, and s is determined by the parameters **detectarclines-wavesol-deg-lambda**, **detectarclines-wavesol-deg-order**, and **detectarclines-wavesol-deg-slit**. For single pinhole exposures, which by definition only have one position along the slit, the degree in s is fixed to zero. The fit is done globally for all lines across the detector, and a sigma-clip algorithm, controlled by the parameters **detectarclines-clip-sigma**, **detectarclines-clip-niter**, and **detectarclines-clip-frac**, is applied during the fit to remove outlier line identifications.

11.15 Order tracing via detection of continuum on pinhole flats (xsh_orderpos)

For each order, the order trace along the dispersion direction is sampled every **detectcontinuum-ordertab-step-y** pixels in the Y direction, and the predicted X position of the trace is calculated using the polynomial model stored in the guess order table ORDER_TAB_GUESS_ARM. Then, for each (X,Y) coordinate pair, a 1D window in the X-direction of half-size **detectcontinuum-search-win-hsize** pixels is extracted from the continuum pinhole flat image and filtered by a running box-car median of half-size **detectcontinuum-running-win-hsize** in order to remove cosmic ray hits from the image window. If the parameter **detectcontinuum-running-win-hsize** is set to zero, then this median filtering is not performed. An initial estimate of the detected trace position in the image window is taken as the pixel with the maximum value in this filtered window. This new position is used to extract a new image sub-window of half-size **detectcontinuum-fit-win-hsize** from the original unfiltered image, and a refined trace position is determined via a 1D Gaussian fit.

The set of (X,Y) coordinate pairs that trace the orders on the continuum pinhole flat image are fit using a polynomial in Y of degree **detectcontinuum-ordertab-deg-y**, and then any residuals that are worse than **detectcontinuum-clip-res-max** pixels are rejected. After this first fit, the remaining (X,Y) coordinate pairs are fitted again with the polynomial in Y and the fit is further iterated using a sigma-clip algorithm controlled by the parameters **detectcontinuum-clip-sigma**, **detectcontinuum-clip-niter**, and **detectcontinuum-clip-frac**.

A similar polynomial fitting procedure is used in xsh_predict to generate a guess order table.

11.16 Detection of the order edges on the master flat (xsh_mflat)

The input order table (ORDER_TAB_CENTR_ARM) is used to locate the central order traces, and each order is divided into chunks of size $2 \times$ **detectorder-chunk-half-size** $+1$ in the y-direction (i.e. the dispersion direction). In each chunk, the pixels are collapsed to obtain a 1D profile in the cross dispersion direction with enhanced S/N. If the S/N at the central pixel is below the threshold **detectorder-min-sn**, then the profile is rejected from the edges computation. In each 1D profile, the edges on either side of an order are considered detected when the flux drops to below a fraction **detectorder-edges-flux-thresh** of the central flux. If the resulting order width is less than **detectorder-min-order-size**, then the profile is rejected from the edges computation. Finally, for each edge, the set of (X, Y) coordinates that trace the edges are fit with a polynomial in Y of degree 5.

11.17 Detection of IFU flat slices traces (xsh_mflat)

IFU edges are determined by either dividing the X flat edge inter-distance by 3 (if **slice-trace-method** is set to'fixed'), or applying the “Sobel” or “Scharr” edge detection methods. The IFU slices edges are determined by a Gaussian fit over a window of 5 pix centred on the expected position (as determined by the method 'fixed'). Then the Gaussian centres are fit by a polynomial.

11.18 Master Flat creation (xsh_mflat)

This recipe generates a master flat frame from a set of flats by performing the following data reduction steps:

1. Corrects the flat frames for bias level and dark current.
2. Determines the exposure level of each flat frame by determining for each order the mean exposure level over ten sampling windows uniformly distributed along the order and then taking the mean of these values over the orders. Each flat frame is then normalised by its exposure level, and the normalised flat frames are combined into an initial master flat frame by median combining.
3. Divides the initial master flat frame from (2) into each calibrated flat frame from (1). Now each and every pixel in the resulting flat-fielded flat frame gives an estimate of the exposure level of the flat frame, regardless of the order it belongs to. Then, to estimate the exposure level of each flat frame, the recipe calculates the median of all pixels in all orders on the flat-fielded flat frame.
4. Normalises each calibrated flat frame from (1) by the exposure levels derived in (3), and combines the flats using the median to obtain the final master flat frame.
5. The inter-order background on the master flat frame is determined and subtracted (see next Section).
6. Low sensitivity pixels (due to dust etc.) are detected on the master flat frame and flagged in the bad pixel mask. This is done by flagging pixels (code 16384) with absolute residuals greater than 20σ on the difference image produced by subtracting a median smoothed (7×7 pixel box) version of the master flat from itself.

11.19 Interorder background determination (xsh_mflat; stare recipes)

The algorithm for fitting the inter-order background starts by using the detected order edges on the master flat frame to determine the bisecting traces of the inter-order regions along the dispersion direction (which are all nearly parallel to the image y-axis). Each bisecting trace is then used to define an inter-order region that has curved right and left x-coordinate limits defined by shifting the trace by M pixels and by $-M$ pixels along the detector x-axis, respectively, and that has upper and lower y-coordinate limits matching the ends of the shorter of the two bounding orders. The quantity M is then an algorithm parameter (**background-edges-margin**) specifying the half-width of each inter-order region. A further two image regions, one to the right of the right-most order and one to the left of the left-most order, each with appropriate margins, are added to the set of inter-order regions. The algorithm then fits a two-dimensional polynomial surface to the pixel values belonging to the set of inter-order regions, ignoring bad pixels and using optimal inverse-variance pixel weights. The fitting procedure is iterated to allow the removal of outlier pixel values through sigma-clipping and the iterations stop when no more pixels are rejected. The degree of the polynomial surface to be fit is defined by **background-poly-deg-x** and **background-poly-deg-y**, and the threshold for sigma-clipping is defined by **background-poly-deg-kappa**.

11.20 Instrument response determination

Initially the flux table corresponding to the reference order merged spectrum of the observed standard star is extracted from an input catalogue, first interpolated to the same steps as the observed spectrum. We correct the wavelength scale of the model spectrum to the same radial velocity as the observed spectrum. For VIS and NIR data we apply a correction of the telluric absorption by looking for the best fitting spectrum within a catalog of telluric model spectra. Since the model spectra are provided for a large range of water vapur content, but only for few values of CO_2, we mask CO_2 regions when fitting the response later. The response is computed by dividing this corrected model spectrum of the standard star (in erg cm^{-2}s^{-1}Å^{-1}) by the 1D extracted observed spectrum of the standard star corrected for gain, exposure time, atmospheric extinction (the atmospheric extinction table is interpolated to get the same binning) and, in the case of VIS and NIR data, for telluric absorption. To reduce the noise of the resulting ratio spectrum we apply a median filter of eleven pixels half width. Then we apply a cubic-spline fit to the points defined in RESP_FIT_POINTS_CAT_ARM to get the final response. Regions of very high telluric absorption (between J and H and between H and K in the NIR arm) or CO_2 absorption (see above) are masked and interpolated.

The response is obtained with the following equation:

$$\mathrm{Response} = \frac{\mathrm{STD_{fluxtable}} \times \mathrm{exptime} \times \mathrm{gain[ADU/e-]}}{\mathrm{STD_{observed}} \times 10^{(0.4 \cdot \mathrm{airmass} \cdot \mathrm{ext})}} \tag{2}$$

The resulting response function is in erg cm^{-2} electron^{-1}. The response is derived from the order merged flux standard spectrum and applied on the merged science spectrum.

11.20.1 Flux calibration

If the user provides the instrument response and the atmospheric exctiontion tables in input of a science recipe, the merged 2D and 1D spectra are then flux calibrated. This operation is performed by first dividing the observed spectra by exposure time and the detector gain and correction for atmospheric extinction, and then multiplying those by the instrument response. The resulting spectrum is thus flux-calibrated, in units of erg cm^{-2}Å^{-1}.

11.21 Efficiency determination

The efficiency at a given wavelength λ is computed as:

$$\epsilon(\lambda) = \frac{I_{STD}^{XSH}(\lambda) \cdot 10^{0.4 \cdot ext(\lambda) \cdot (airp - airm)} \cdot gain \cdot E_{phot}(\lambda)}{T_{exp} \cdot A_{tel} \cdot I_{STD}^{ref}(\lambda)} \cdot factor$$

where $I_{STD}^{XSH}(\lambda)$ is the extracted standard star spectrum as observed by X-Shooter, corrected for the contribution from the sky background, at a given wavelength λ, $ext(\lambda)$ is the atmospheric extinction coefficient, $airm$ is the airmass at which the standard star was actually observed, $airp$ is a parameter to indicate if the efficiency is computed at airmass=0 (no atmosphere) or at a given value (usually the one at which the reference standard star spectrum may be tabulated). $gain$ indicates the detector's gain, and $E_{phot}(\lambda)$ is the energy of one photon ($E_{phot}(\lambda) = \frac{10^7}{\lambda \cdot 1.986 \cdot 10^{19}} \cdot J \cdot um^{-1}$), T_{exp} is the total exposure time in seconds, A_{Tel} is the UT telescope collecting area ($51.2 \cdot 10^4 \cdot cm^2$), $I_{STD}^{ref}(\lambda)$ is the flux calibrated spectrum of the reference source. $factor$ is a multiplicative number that corrects for the fact that in the previous formula some quantity has been expressed in different units.

11.22 Order rectification

The process of resampling the spectral orders from image space to wavelength-slit space is referred to as rectification. The sampling step along the wavelength and slit directions is defined by the parameters **rectify-bin-lambda** and **rectify-bin-slit**. The exact values of the order limits are adjusted so that the grids of overlapping orders can be superimposed without further resampling.

The resampling works as follows. For each pixel in each order in the wavelength-slit space, the corresponding pixel position in the original image is derived using the wavelength solution. A kernel is then used to convolve the original image at the required pixel position to derive the corresponding pixel value in the wavelength-slit space. The kernel is defined by the parameters **rectify-kernel** and **rectify-radius**. The pixel uncertainties are propagated in the proper analytical fashion and the pixel flags are also propagated within the kernel radius.

Note that this resampling is not equivalent to a rebinning of the spectrum because the kernel radius is the same in the original image space regardless of the sampling steps chosen in the wavelength-slit space. Hence there is no advantage in terms of S/N in choosing a coarse sampling grid in wavelength-slit space. Any binning of the output spectra should be done as a further step and this is not supported by the XSHOOTER pipeline.

The resampling process introduces correlated noise into the two-dimensional rectified images which sometimes manifests as ripples in the spectrum. Standard extraction across the slit mitigates this effect somewhat. However it is still possible to see such ripples in the output 1D extracted spectra.

11.23 Cosmic rays flagging on individual frames

The method used to flag cosmic rays (as opposed to removing them) is that of Laplacian edge detection as specified by van Dokkum (2001). We refer the reader to the van Dokkum paper for more details on the algorithm. The controlling parameters are **removecrhsingle-sigmalim**, **removecrhsingle-flim**, and **removecrhsingle-niter**. To increase/reduce the number of detected cosmic rays, one can increase/reduce the values of the parameters **removecrhsingle-sigmalim** or **removecrhsingle-flim**.

11.24 Sky subtraction for stare observations

Sky subtraction is performed by default (**sky-subtract = TRUE**) for stare observations in the recipes xsh_respon_slit_stare and xsh_scired_slit_stare. To estimate the sky spectrum, the pixel values whose slit position lies outside of the object mask and that are far enough away from the slit edges (as specified by **sky-slit-edges-mask** in ″) are tabulated as a function of wavelength (taken from the wavelength map). The sky regions may alternatively be defined by the parameters **sky-position1**, **sky-hheight1**, **sky-position2** and **sky-hheight2**. The resulting 1D sky spectrum is then smoothed using one of the following methods (**sky-method**):

- The **MEDIAN** method uses a running median of box half-size **sky-median-hsize**.
- The **BSPLINE** method is controlled by the following parameters: **sky-bspline-nbkpts-first** and **sky-bspline-nbkpts-second** (number of spline break points for the first and second sky subtractions, respectively), **sky-bspline-order**, **sky-bspline-niter**, **sky-bspline-kappa** and **sky-bspline-sampling**.

The 1D sky spectrum, along with the wavelength map, is used to create a 2D sky frame that can be subtracted from the 2D science spectrum.

11.25 Object localization

The localization of an object is defined by three polynomial expressions giving the centroid position on the slit in ″, as well as the object slit limits, as a function of wavelength. The localization is determined on the rectified merged orders via three different methods:

- When **localize-method**=MANUAL, the localization can be fixed by the user by setting the central position **localize-slit-position** and the half size **localize-slit-hheight**.
- When **localize-method**=MAXIMUM, the merged 2D spectrum is chopped into chunks of length **localize-chunk-nb** and the signal is collapsed along the wavelength direction to give a 1D profile. The centre is determined as the position of maximum flux and the edges are where the flux is a fraction **localize-thresh** of the central flux.
- When **localize-method**=GAUSSIAN, the merged 2D spectrum is processed in the same way as for the MAXIMUM option. However, the 1D profile is fit with a Gaussian. The centre is taken as the Gaussian peak and the edges are defined at the 3-sigma distance.

For the MAXIMUM and GAUSSIAN methods, the centre and edge positions are fit with a polynomial of degree **localize-deg-lambda** using sigma-clipping with parameters (**localize-kappa** and **localize-niter**.

11.26 Standard extraction

Standard extraction by integration over the object aperture is carried out on the 2D rectified orders before they are merged. Bad pixels in the extraction aperture at a particular wavelength trigger the construction of a local spatial profile using all wavelength bins with no bad pixels within **stdextract-interp-hsize** wavelength bins of the current wavelength bin. The local profile is then optimally scaled to match the good pixels at the current wavelength and the scaled profile is integrated to provide the flux estimate at the current wavelength. For parts of the 2D spectrum with many bad pixels in consecutive wavelength bins, it may be necessary to increase the value of **stdextract-interp-hsize** to avoid "gaps" appearing in the output 1D spectrum.

Note that the pixel flag decodification into good and bad pixels may be controlled by the parameter **decode-bp**. This is important if you want to include or exclude certain types of pixels from the standard extraction integration/interpolation (e.g. cosmic rays).

11.27 Optimal extraction

The recipes xsh_respon_slit_stare and xsh_scired_slit_stare offer the additional option of using an "optimal extraction". Note that this option is not offered for the offset and nod recipes. The code for the optimal extraction is written by Goldoni et al. (ADASS XXI, 2012, ASP Conf. Ser. Vol. 461, Page 741) and does not include any treatment of bad pixels. The code is distributed as provided by the consortium, and the corresponding algorithm and its implementation have not been investigated in detail.

Testing by ESO indicates that for point sources and default parameters, the stare recipes deliver spectra at selected wavelength regions in the UVB arm with noise $\sim$10-20% lower than the standard extraction for point sources. However, between these clear regions, significant artifacts are introduced. The optimal extracted spectra also differ in flux scale by several hundred percent from the standard extraction. Master response curve therefore does not apply to the optimally extracted spectrum. The performance is illustrated in Figure 11.1.

11.28 Reduction of STD star frames in SLIT configuration - stare/offset/nod

The same data reduction steps up to the computation of the merged spectrum are performed as for the corresponding science recipe. Then if the observed standard star is one listed in the reference flux standards catalogue (see 8.6), the spectrum response is computed (see 11.20). and it is applied to the order merged 2D and 1D spectra of the flux standard after having corrected this by atmospheric extinction, exposure time and gain, generating the flux calibrated 2D and 1D spectra in units of erg cm^{-2}s^{-1}Å^{-1}.

11.29 Science reduction in SLIT stare mode (xsh_scired_slit_stare)

The steps performed by this recipe are as follows:

1. The input is N science frames which must all have the same exposure time.
2. Prepares the science frame(s) in PRE format.
3. Performs optional master bias and master dark correction on each science frame.

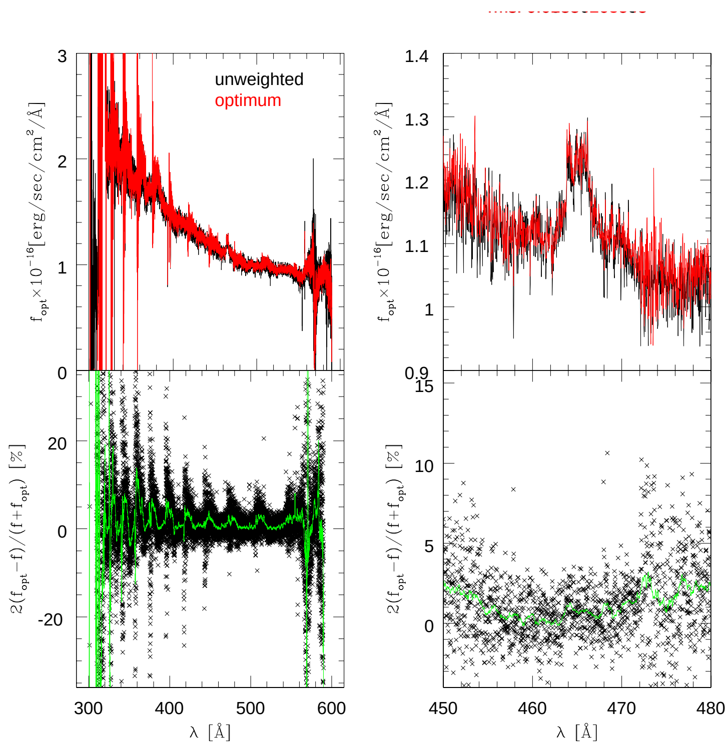


Figure 11.1: Comparison of extracted spectrum of a point source. The upper panels show the whole UVB spectrum (left) and a selected wavelength region around a spectral line (right). The spectrum from standard extraction is shown with black lines. The spectrum from optimal extraction has been scaled in flux and is shown in red. The noise as measured close to the line is ∼15% lower with optimal extraction. The lower panels directly compare the two spectra. Black points show the difference of the two spectra scaled by the mean. The green lines are the binned data. It can be seen that for about 50 percent of the wavelength range, the two extraction methods differ significantly from each other. The data used in this plot were kindly provided by Oliwia Madej (O.K. Madej et al. 2012, arXiv:1212.0862).

4. Detects cosmic rays on each science frame using Laplacian edge detection.

5. If there is more than one input science frame, then they are combined into a single frame using the median (**stack-method = median**) or sigma-clipped mean (**stack-method = mean**).

6. Determines and subtracts the inter-order background from the combined science frame.

7. Divides the master flat frame into the combined science frame.

8. If **sky-subtract = TRUE**, then the recipe determines and subtracts the sky background from the combined science frame.

9. Rectifies the combined science frame on an order by order basis.

10. Uses standard extraction to extract the spectrum on the rectified orders.

11. Merges the 1D (and 2D) spectra into a single spectrum.
12. Flux calibrates the 1D (and 2D) spectra.

11.30 Science reduction in SLIT offset mode (xsh_scired_slit_offset)

The steps performed by this recipe are as follows:

1. The input is N science and N sky frames which must all have the same exposure time.
2. Prepares the science and sky frames in PRE format.
3. Detects cosmic rays on each science and sky frame using Laplacian edge detection.
4. Orders the science and sky frames by date of observation and subtracts the sky frames from the science frames in pairs.
5. Divides the master flat frame into each sky-subtracted science frame.
6. Rectifies each sky-subtracted science frame on an order by order basis.
7. Combines the rectified sky-subtracted science frames into a single frame using the median (**combinenod-method = MEDIAN**) or the mean (**combinenod-method = MEAN**).
8. Uses standard extraction to extract the spectrum on the rectified orders.
9. Merges the 1D (and 2D) spectra into a single spectrum.
10. Flux calibrates the 1D (and 2D) spectra.

11.31 Science reduction in SLIT nodding mode (xsh_scired_slit_nod)

The steps performed by this recipe are as follows:

1. The input is N science frames (N must be even) which must all have the same exposure time.
2. Prepares the science frames in PRE format.
3. Detects cosmic rays on each science frame using Laplacian edge detection.
4. Orders the science frames by date of observation.
5. For a sequence such as AAA BBB CCC DDD, the recipe combines the science frames at the same position into the corresponding sequence A B C D. The combination is performed using the median (**stack-method = median**) or sigma-clipped mean (**stack-method = mean**).
6. Subtracts the pairs of (combined) nodded observations, e.g. A-B, C-D, etc.
7. Divides the master flat frame into each subtracted science frame pair.

8. Rectifies each subtracted science frame pair on an order by order basis.
9. If **correct-sky-by-median** is set to **TRUE**, then the recipe calculates and subtracts the median pixel value for each column (wavelength) in the rectified frame from the column pixel values.
10. For each subtracted science frame pair, the recipe forms [(A-B) - shifted(B-A)].
11. Combines the set of frames of the form [(A-B) - shifted(B-A)] using the median (**combinenod-method = MEDIAN**) or the mean (**combinenod-method = MEAN**).
12. Uses standard extraction to extract the spectrum on the rectified orders.
13. Merges the 1D (and 2D) spectra into a single spectrum.
14. Flux calibrates the 1D (and 2D) spectra.

11.32 IFU object traces determination

The cube spatial and wavelength sampling steps are set according to the observing wavelength range (arm). Using the information on the order centre and IFU slices edges trace locations, as contained in the IFU edge order table, for any table row (in this way scanning each Y of all detected orders), for each IFU slice, it is performed a Gaussian fit (eventually degenerate to a centroid fit) of the object X position in the pre-processed science frame, constrained to X range determined by the predefined IFU slices edge positions. In this way the object traces are determined in each slice of the IFU. Then each (x, y) pair found in each IFU slice is converted in corresponding (s, λ) positions using the wave and slit map information (poly mode) or the model (physical model mode).

The distribution of $s^{upp}(\lambda)$, $s^{cen}(\lambda)$, $s^{low}(\lambda)$ is different, but we know that the sum $s^{upp}(\lambda) + s^{cen}(\lambda)$, and $s^{low}(\lambda) + s^{cen}(\lambda)$, is a constant, equal to twice the distance between the slices from the optical center.

Moreover it is known that should be valid the following relations:

$$s^{upp} = \sigma \cdot s^{cent} + c_0^{upp} \tag{3}$$

$$s^{low} = \sigma \cdot s^{cent} + c_0^{low} \tag{4}$$

where $\sigma = -1$, was determined in the lab.

Using the s^{upp} and s^{low} point distributions determined for several standard stars we have determined the best fit values c_0^{upp} and c_0^{low}.

More accurate relations that take into account of a small wavelength dependence are:

$$s^{upp} = -s_{cent} + c_0^{upp} + c_1^{upp} \cdot \lambda + c_2^{upp} \cdot \lambda^2 \tag{5}$$

$$s^{low} = -s_{cent} + c_0^{low} + c_1^{low} \cdot \lambda + c_2^{low} \cdot \lambda^2 \tag{6}$$

The coefficients c_i^{upp}, c_i^{low} $(i = 0, 1, 2)$ have been determined minimizing residuals in the final cube reconstruction over a set of standard stars (see Sec. 11.33).

11.33 IFU 3D cube generation

The cube spatial and wavelength sampling steps are set according to the observing wavelength range (arm). For each detected order m, for each detected wavelength λ, for each slit value expected in the central IFU slice s^{cen}, the (x^{cen}, y^{cen}) position corresponding to the current triplet (m, λ, s^{cen}) is determined, using either the flexure corrected wavelength solution obtained in poly mode or the physical model configuration previously optimized first on single-pinhole (format-check) and multi-pinhole (2dmap) frames, and corrected for flexures. The (x^{upp}, y^{upp}) and (x^{low}, y^{low}) corresponding positions are determined using the relations 5 and 6 obtained as described in 11.32. The corresponding fluxes, errors, and qualifier are determined by interpolating the pre-processed IFU science frame (and corresponding error) images with a kernel of a given **rectify-radius** and **rectify-profile**. Those values are stored in the cube position corresponding to the given IFU slice, spatial and wavelength coordinate. Cube values points corresponding to wavelength where differents orders overlaps are averaged. For quality control are computed the object traces in each IFU slice of the merged cube using a Gaussian fit (eventually degenerating to a centroid determination) at each wavelength.

The reconstructed cube, its error and qualifier and the object traces are the products of this algorithm.

11.33.1 IFU trace position determination — staring mode (xsh_geom_ifu)

The recipe is aimed at tracing the position of a pointlike object in the IFU and derive shifts along the slitlets as a function of wavelength with respect to a reference position. These shifts will correct any distorsion due the atmospheric dispersion, the wavelength solution and the optics of the IFU.

The tracing consists in an iterative localization of the standard star in each of the IFU slitlets:

1. The first steps of the reduction are similar to other recipes. One additional option is offered: the flat-fielding can be done using a model of the blaze function (**flat-method**=blaze) instead of the master flat-field (**flat-method**=master). Using the blaze function allows to produce cleaner edges of the slitlets.

2. The slitlet is rectified and the different échelle orders are merged together.

3. The full wavelength interval is divided into chunks (whose half size is defined by **localizeifu-chunk-hsize**). The positions around sky emission lines can be masked by setting **localizeifu-use-skymask**. The slit coverage can be truncated in order to mask the edges of the slit. The lengths in $''$ to be masked are defined by **localizeifu-slitlow-edges-mask** and **localizeifu-slitup-edges-mask**. For each chunk, a median filter (**localizeifu-smooth-hsize**) is applied on the summed cross-dispersion profile and a Gaussian fit is done to locate the object at the given central wavelength. This Gaussian expression has a polynomial component to fit the background. The degree of this polynomial expression is to be ≤ 2 and is controlled by **localizeifu-bckg-deg**.

4. The results of the Gaussian fits are cleaned by removing outliers in FWHM and SNR (**localizeifu-sigma-low** and **localizeifu-sigma-up** are the limits for σ, **localizeifu-snr-low** and **localizeifu-snr-up** are the limits for the SNR).

5. The Gaussian localization vector is interpolated to provide a localization at each wavelength, even where the fit failed.

6. This localization vector is filtered by a wavelet à trous transform, decomposed in a series of **localizeifu-wavelet-nscales** frequencies ; the highest ones (**localizeifu-wavelet-hf-skip**) are skipped in order to decrease the noise in the localization.

7. The reference position is derived as the median localization around the reference wavelength (in the wavelength interval **correctifu-lambdaref** $\pm$ **correctifu-lambdaref-hsize**) ; the shift vector is computed as the difference between the localization vector and the reference position.

8. The total shift vector is incremented with the derived shifts.

9. If the maximum number of iterations **correctifu-niter** is not reached, the slitlet is *shifted and* rectified again (using the derived shifts) and goto 3.

The total shifts are stored in three different tables (one per slitlet). If **do-cube** is set to TRUE, a datacube is produced as quality control. In this final datacube, the localization of the standard star is straightened.

11.33.2 Science reduction in IFU configuration — staring mode (xsh_scired_ifu_stare_drl)

The recipe is aimed at reducing SCIENCE data observed in IFU staring mode.

If three or more science frames are present in the set-of-frames, they are combined into one median frame to reject outliers. The master bias frame is subtracted (UVB and VIS arms, not for NIR arm). The master dark frame is subtracted (In NIR, the OFF frame is subtracted).

The order table is used to locate the inter-order regions where the background is estimated on a grid along the dispersion direction (parametrized by **background-nb-y**). At each point of this grid, the background is estimated indide a box (defined by **background-radius-x** and **background-radius-y**) as the median or the minimum value (parametrized by **background-method**). The background frame is then built as a 1D-spline of the estimated values and is smoothed (**background-smooth-x** and **background-smooth-y**). The background is then subtracted from the data.

The sky background is *not* removed from an IFU in staring mode.

If only one science frame was given in input, the cosmic ray hits are corrected using the Van Dokkum algorithm.

Each slitlet is rectified from the pixel space (X, Y) onto a regular grid in the (λ, s) space. The sampling of this grid is defined by **rectify-bin-lambda** and **rectify-bin-slit**. The rectification method uses the wavelength solution provided by the user (either by the physical model or the 2D polynomial solution) for each slitlet to derive X and Y for the grid of (λ, s). The positions on the slit are corrected from the offset tables produced with xsh_geom_ifu, OFFSET_TAB_CEN/DOWN/UP_IFU_ARM. These shifts correct for the atmospheric dispersion and possible distorsions of the global wavelength solution.

Then the rectification uses the kernel and its radius defined by **rectify-kernel** and **rectify-radius** respectively to estimate the flux at the *shifted* positions of the grid. If the frame was not divided by the flat-field, the flux is conserved with the option **rectify-conserve-flux**.

The localization is done on the rectified slitlets to derived the position of the slit. This position is a polynomial expression function of the wavelength (with a degree **localize-deg-lambda**). The rectified frame is collapsed in several chuncks and a 1D Gaussian fit on the collapsed chunk gives the position. The positions are then fitted by the polynomial expression. From this position, the edges are derived and parametrized by the threshold **localize-thresh**. This automatic localization can be overriden by a manual one (parametrized by **localize-method**) which allows the user to supply the localization and the half height of the object in arcseconds (**localize-slit-position** and **localize-slit-hheight** respectively).

The object bad pixel map is merged with the telluric bad pixel map to mark the positions of telluric absorption features.

The rectified échelle orders are merged into a single 2D spectrum, for each slitlet.

The 3D datacube is constructed.

11.34 Science reduction in IFU configuration — offset mode (xsh_scired_ifu_offset)

The recipe retrieves the input parameters and frames and the raw frames are prepared (see 11.1). Object and Sky observations are separed. For each frame pair the Sky frame is subtracted from the Object one.

The order table is used to locate the inter-order regions where the background is estimated on a grid along the dispersion direction either with a spline or a (default) polynomial method (see 11.19). The background is then subtracted from the data.

The cosmic ray hits are flagged using the Van Dokkum algorithm (see 11.23, van Dokkum method). The frame is divided by the master flat field. If **trace-obj** and **compute-map** are set to TRUE, for quality control, the recipe computes the object traces in each IFU slice (see 11.32). The 3D datacube is constructed (see 11.33).

11.35 Instrument model configuration recover (xsh_cfg_recover)

This recipe allows recovery of the X-Shooter physical model parameter configuration file (hereafter "config") in the case of significative spectral format shifts due to earthquakes or maintenance and upgrade interventions. *It should seldom be necessary to use this recipe and when it is required it should be handled by an instrument scientist*. The process is described in more detail in [3] where there is also some description of how to establish what is the correct default config for science exposures from a given epoch.

Two are the possible scenarios:

- The transformation of the spectral format is known to be a simple linear transformation (with any higher order non-linear effects <5pix everywhere on the detector). In this case **first-anneal**=FALSE and the only **last_step** is performed. The user provides the last valid config file before the format change and the approximate linear x, y translation of the spectral format using the **offx** and **offy** parameters. The recipe modifies the config following the safefit algorithm (section 11.36).
- Large non-linear spectral format shifts. In this case the user needs to provide an additional input table with interactively measured (to +/-1pix) centroids of a 12-16 of prominent spectral features in a calibration lamp exposure taken after the spectral format change occurred. The user will set **first-anneal**=TRUE and the recipe will perform a first optimisation on the input line list and config (**niter** iterations) to improve the instrument configuration model to an accuracy of 1-2 pixels. This is sufficient to allow a longer list of calibration lines to be automatically identified using the physical model (see section 11.36). Note that the

config output by the **first_anneal** step is *only* used for the automatic line identification and centroiding. The subsequent optimisation in **last_step** uses the input config as its starting point.

The main steps performed during the last step are the following:

- The spectral format corresponding to a given input config is determined. Then to correct the initial model prediction:
- If **first_anneal** was skipped then the user should supply the approximate linear x, y translation of the spectral format using the **offx** and **offy** parameters.
- The **method** safefit is used to automatically identify the list of calibration lines and determine their centroids in the post-format-change exposure.
- Finally a predefined subset of the model configuration parameters automatically optimised with the simulated annealing technique to better match the previously selected list of lines. The parameter **anneal_niter** specifies the number of iterations of the annealing process.
- The recipe outputs the best config obtained after annealing. This can be further fine tuned with the xsh_2dmap recipe in order to produce a new default physical model configuration file

11.36 Algorithm description of safefit method implemented in xsh_cfg_recover

To properly work this algorithm requires:

- A model configuration that performs to better than 10pix (preferably $<$ 5 pix) accross the detector. This will usually come from the first_anneal data reduction stage.
- An arc lamp exposure with corresponding line list. Usually this is pen-rays (Ne, Xe, Ar & Hg) for UVB and Th-Ar HCL for VIS and NIR. The line list should only contain lines isolated to within 5 pixels. For Th-Ar customised lists have been provided for UVB, VIS and NIR. For the combined pen-rays so far only the NIR list is available. This should be in the xsh_cfg_recover sof.
- A spectral format table (optional, if not provided it is generated OTF from the config)
- (optional) Master dark and master bias files for preparing the arc lamp exposure.

This method implements the following data reduction algorithm.

- A 1d spectrum for each order is extracted using the physical model and the config suplied as input or from the first_anneal step:
  - Loop over orders
  - Loop over dispersion pixels
  - Extract the flux in a 11 pixel high window centred on the x-dispersion co-ordinate that the model locus predicts for this order and dispersion co-ordinate. This is stored in the 2D array extracted (1st index order; 2nd index dispersion co-ord)

  - The peaks (cent[i]) in this 1d extracted spectrum are identified.

- The peaks (cent[i]) in this 1d extracted spectrum are identified.
- The peaks are matched to wavelengths where possible:
  - Loop over orders
  - Loop over peaks, i
  - Determine which peaks are isolated on the right of each peak:
    * Loop over margin size, w
    * Compute the barycenter, cent_b, of flux in the extracted 1d spectrum in the window from (cent[i]-wmin) to (cent[i]+w), where wmin is the typical line half width.
    * The maximum value of w for which abs(cent[i]-cent_b) is less than 0.5pix is recorded as the isolation on the right of this peak.
  - Determine which peaks are isolated on the left of each peak (analogous to the loop for the right above)
  - If both left and right isolation exceeds 5pix then continue (otherwise go to next peak).
  - Search for a match to this peak in the input line list:
    * Loop over entries in the input line list
    * For each line use the input model config to compute the expected co-ordinates in this order (actually get the value from the tab_xy_guess table computed earlier)
    * If the computed dispersion co-ord matches cent[i] to within 2.5 pix then cent[i] is taken to be the dispersion co-ord for this wavelength. Otherwise disregard this peak and go to the next.
  - Determine the x-dispersion co-ordinate from the polynomial description of the order shapes
- The co-ordinates associated with each wavelength are passed to the annealing as in the other methods.

A Workflow technicalities

This section will present some detailed explanations of the X-shooter Reflex-based workflow.

A.1 Data requirements and optional inputs

In order to run the workflow with a given data set, a number of calibrations and science data must be present. The workflow functions with the concept of science data sets: a set of files that contain a coherent set of science and calibration data.

The minimum coherent data set is composed of:

- A science frame (observed in stare or offset or nod modes).
- A set of bias frames (UVB,VIS).
- A single-pinhole arc lamp frame guess spectral format determination.
- A single-pinhole lamp spectrum to trace the different orders.
- A set of flat frames.
- An multi-pinhole arc lamp frame for determination of the 2D geometry solution.
- Static calibrations: Line reference table and extinction coefficient. table

On top of that, there are some optional calibrations that will be used by the workflow *if* they are present:

- A set of darks.
- A standard star taken the same night for flux calibration purposes.

The conditional execution of the **xsh_mdark** and **xsh_respon_mode (mode=stare, offset or nod)** recipes is implemented in subworkflows `Master Dark Creation` and `Instrument Response`.

The data packages sent by ESO to researchers (originally in form of PI-PACKS or recently CalSelector data sets) contain all the required calibrations to run the workflow. Ideally, it is enough to feed a given data set into the workflow to reduce the data.

For a detailed explanation about how these calibrations are organised and associated, please refer to Section A.3.

A.2 Overall layout

The workflow structure has several parts:

- Top area. Contains annotations about the workflow and the main parameters to setup the workflow. Note that not *all* the parameters of the workflow are set here: each actor has its own parameters that are set individually.
- Bottom area. This contains all the actors that actually execute or perform a workflow action. It is also composed of several parts:
  - Initialisation actors on the left side. These actors prepare the rest of the workflow to start with the data reduction. It includes setup of required intermediate variables, the data organisation, the data selection and the routing of the data. Please note that the data organisation and selection is performed only once, while the routing is performed once for each data set that has been selected.
  - Recipe execution. The middle part of the bottom area contains the actors that execute pipeline recipes. This is basically where the logic of the data reduction chain is implemented.
  - Closing of the reduction for this data set. The right side of the workflow contains actors that perform the last actions needed after the reduction of a data set. This includes renaming of the final science data, an interactive data display (disabled by default) and some housekeeping of variables.
- Subworkflows. Some of the actors in the workflow are actually subworkflows. That means that they contain another workflow inside. To inspect the subworkflow, right-click on it, and select `Open Actor`.

A.3 OCA rules

The OCA rules are the mechanism used to Organise, Classify and Associate the data. These rules are stored in a human-readable file which contains several sections, one for each of the required tasks.

The DataOrganiser is the component that makes use of these rules. The parameter `Oca File` specifies the proper OCA rule file to use. It is advised to check that it points to the right location (although the installation procedure should take care of that). Figure A.1 shows the parameter to inspect to check that the OCA rules are ok.

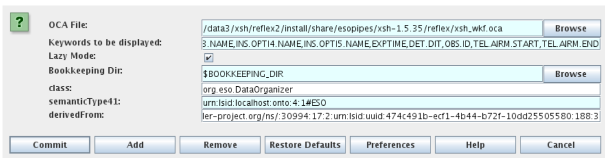


Figure A.1: *Parameter to change for the OCA file, used for classification, grouping and association of data.*

Here we present a brief summary of the meaning of the current OCA rules provided with the workflow:

- The raw data is classified according to the DPR keywords.
- The products of the recipes are classified according to the PRO.CATG keyword.
- All X-shooter data are grouped by arm.
- The raw biases are grouped by detector read-out mode and observation template.

- The raw darks are grouped by observation template and exposure time (UVB/VIS) or DIT (NIR).
- The raw flats are grouped by observation template, slit-id, detector read-out mode (UVB/VIS) or DIT (NIR).
- The raw format check, wave and order-definition data are grouped by observation template detector read-out mode (UVB/VIS) or DIT (NIR).
- The standard star and science data are grouped by observation template, slit-id, detector read-out mode (UVB/VIS) or DIT (NIR).
- All the calibrations are directly associated to the science data. If several groups of calibration data match the rules, then usually the closest in time is chosen.
- The raw biases and raw darks are associated if the detector read-out mode matches (UVB/VIS).
- The raw darks are associated if the detector DIT matches (NIR).
- The raw flats are associated if the detector read-out mode (UVB/VIS) and slit-id match.
- The format check, wave, and the order definition data are associated if the arm matches.
- The standard star and master response are associated if the arm matches.

A.4 Workflow fine tuning and hints

We have collected several hints to fine tune and exploit all the capabilities of the workflow.

- Check disk space before starting to reduce the data. The reduction of all the demo data sets will need at least 1 GB. However, at least 5 GB are recommended. This disk space requirement applies to directories pointed to by the `TMP_PRODUCTS` and `END_PRODUCTS_DIR` variables.
- If the user would like to use the stare data reduction strategy for nodding or offset data select the OCA rules file xsh_wkf_stare.oca instead of the default xsh_wkf.oca from the Data Organiser actor configuration. In case of NIR data the user must provide also raw dark frames with the appropriate DIT.
- The Flat Strategy actor allows the user to use switch between two strategies to select a flat field for the flux calibration reduction. 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 of the standard observations.
- All the intermediate products created by the workflow are stored in `TMP_PRODUCTS_DIR` directory. The subdirectory structure is shown in Figure A.2. For each recipe instance there is a subdirectory, and inside this there is another subdirectory with the timestamp of the execution time.
- The ProductRenamer can be setup to create the desired filenames. Use the `Rename keywords` parameter to change the renaming scheme. It is possible to use keywords from the the header or literal strings (quoted by ”). Currently, there is only one ocurrence in the workflow of the ProductRenamer, just after the science recipe. However, this actor can be placed in several places in the workflow (for instance, to store in the final directory the master flat).

```
├── xsh_2dmap_1
│   ├── 2012-12-20T15:44:36.049
│   ├── 2012-12-20T16:08:44.325
│   └── 2012-12-20T16:29:19.500
├── xsh_flexcomp_1
│   ├── 2012-12-20T15:57:39.193
│   ├── 2012-12-20T16:02:27.042
│   ├── 2012-12-20T16:11:56.721
│   ├── 2012-12-20T16:19:52.704
│   ├── 2012-12-20T16:36:07.900
│   └── 2012-12-20T16:47:47.974
├── xsh_mbias_1
│   ├── 2012-12-20T15:35:44.536
│   ├── 2012-12-20T15:36:32.051
│   ├── 2012-12-20T15:36:54.690
│   ├── 2012-12-20T15:37:35.550
│   ├── 2012-12-20T16:23:28.642
│   ├── 2012-12-20T16:24:00.282
│   ├── 2012-12-20T16:24:31.699
│   ├── 2012-12-20T16:25:00.496
│   └── 2012-12-20T16:39:43.679
├── xsh_mdark_1
│   ├── 2012-12-20T16:05:28.271
│   ├── 2012-12-20T16:15:05.560
│   └── 2012-12-20T16:15:22.717
├── xsh_mflat_1
│   ├── 2012-12-20T15:40:35.336
│   ├── 2012-12-20T15:42:11.036
│   ├── 2012-12-20T15:42:54.848
│   ├── 2012-12-20T16:07:42.900
│   ├── 2012-12-20T16:08:15.443
│   ├── 2012-12-20T16:15:41.652
│   ├── 2012-12-20T16:27:47.684
│   ├── 2012-12-20T16:28:30.942
│   ├── 2012-12-20T16:40:42.561
│   └── 2012-12-20T16:41:28.816
├── xsh_orderpos_1
│   ├── 2012-12-20T15:40:29.703
│   ├── 2012-12-20T16:07:35.325
│   ├── 2012-12-20T16:07:39.071
│   └── 2012-12-20T16:27:38.012
├── xsh_predict_1
│   ├── 2012-12-20T15:38:08.855
│   ├── 2012-12-20T16:05:45.824
│   ├── 2012-12-20T16:06:42.902
│   └── 2012-12-20T16:26:02.981
├── xsh_respon_slit_offset_1
│   ├── 2012-12-20T15:53:56.523
│   ├── 2012-12-20T16:16:16.639
│   └── 2012-12-20T16:43:35.487
├── xsh_scired_slit_nod_1
│   └── 2012-12-20T16:20:48.131
├── xsh_scired_slit_offset_1
│   └── 2012-12-20T15:58:06.740
└── xsh_scired_slit_stare_1
    ├── 2012-12-20T16:02:52.891
    ├── 2012-12-20T16:12:53.574
    ├── 2012-12-20T16:37:03.702
    └── 2012-12-20T16:48:43.920

111 directories, 278 files
amodigli@pc014007$
```

Figure A.2: *Structure of the* `REFLEX_PRODUCTS` *directory.*

- Should the user find data reduction problems, the recipe parameter **debug** allows to increase recipe verbosity. Change this parameter for the recipe of interest.

B Test scripts for quality control

This release includes scripts useful for verifying the quality of the results. These scripts are not compiled during installation. To compile them the user should do the following. Assuming the kit was unpacked under the directory **xsh_kit**:

cd xsh_kit/xsh-kit-2.2.0/xsh-2.2.0
make check
export TDIR=xsh_kit/xsh-kit-2.2.0/xsh-2.2.0/xsh/tests

The user may obtain on-line help for any test script by executing the script without arguments as:

$TDIR/test_xsh_scriptname

Useful scripts are:

- **test_xsh_prepare** This script converts frames from RAW to PRE format.
- **test_xsh_model** To generate theoretical tables corresponding to a given model configuration file, for example after model optimization (performed in xsh_predict, xsh_2dmap).
- **test_xsh_the_map** This script generates a region file associated to a given input THE map (THEO_TAB_SING_ARM or THEO_TAB_MULT_ARM). It may be used with ds9 to project the solution given by a "THE" map (corresponding to a given model configuration, optimized or not) onto an image frame in PRE format.
- **test_xsh_data_order** This script generates a region file to compare with ds9 order traces with corresponding order frames in PRE format.
- **test_xsh_detect_arclines**

  This script may be used, in poly mode, to play with the parameters affecting the line detection, without running the entire recipe that applies it (xsh_predict, xsh_2dmap, xsh_wavecal).
- **test_xsh_subtract_background** This test script may be used to control the results of the spline inter-order background subtraction without executing the full recipe where this algorithm is applied (xsh_mflat, response recipes, scired recipes).
- **test_xsh_resid_tab**. This script applied to residual table products from xsh_predict or xsh_2dmap generates a region file that, loaded together with a single or multi pinhole arc lamp frame in PRE format, can be used to monitor the accuracy of line detection along the spectral format.
- **test_xsh_detect_order** This script may be used to check the accuracy of the detection of the order edges on a flat frame without running the xsh_mflat recipe.
- **test_xsh_data_dispersol** This may be useful, in poly mode, to generate a wave map and a slit map corresponding to a given dispersion file.
- **test_xsh_data_wave_tab_2d**. This script may be used to monitor the accuracy of the order rectification solution in poly mode, or to get for each order the values of WMIN/WMAX to be used in the spectral format table to control the extraction limits of each order. This is possible by loading with ds9 the sky subtracted science 2D image frame together with the region file generated by this script.

- **test_xsh_data_wavemap** This script, for poly mode, determines the wavelength solution along the order centre traces.
- **test_xsh_rectify** This script may be used to test the order resampling without running the response or science reduction recipes.

Examples of usage of these scripts are shown in Section 10.

C Installation

This chapter gives instructions on how to obtain, build and install the X-SHOOTER pipeline. Even if this chapter is kept as up-to-date as possible, it may not be fully applicable to a particular release. This might especially happen for patch releases. One is therefore advised to read the installation instructions delivered with the X-SHOOTER pipeline distribution kit. These release-specific instructions can be found in the file `README` located in the top-level directory of the unpacked X-SHOOTER pipeline source tree. The supported platforms are listed in Section C.1. It is recommended reading through Section C.3 before starting the installation.

A bundled version of the X-SHOOTER pipeline with all the required tools and an installer script is available from `www.eso.org/pipelines`.

C.1 Supported platforms

The X-SHOOTER pipeline has been verified to install and execute correctly with EsoRex on the VLT target platforms:

- Intel Xeon, Core(TM)2, using Linux, and gcc 4.4.1.

and on

- Mac Darwin 12.2.0 (compiling with CC=gcc)

using the GNU C compiler (version 3.2 or newer). Correct execution using EsoRex has been verified on

- Linux Fedora core 11 (with gcc4.4.1),15 (with gcc 4.6.3), 16 (with gcc4.6.3), 17 (with gcc4.7.2)
- Scientific Linux 5.5
- Ubuntu 10.04.4
- Mac Darwin 12.2.0 (compiling with CC=gcc)

C.2 Requirements

To compile and install the X-SHOOTER pipeline one needs:

- the GNU C compiler (version 4.1 or later),
- the GNU `gzip` data compression program,
- a version of the `tar` file-archiving program and
- the GNU `make` utility.

- Processor(s) with 32 Bit architecture. We could verify portability also on Mac snow Leopard 64 and Intel 64 bit platform. Installation of this kit release should be done with the install_pipeline script part of the kit. Installation of this kit together with other ESO pipeline kit distribution (via the script install_pipelinekit) is not yet supported.

For Gasgano support one needs in addition

- the Java Development Kit (version 1.6)

C.3 Building the X-SHOOTER pipeline

The X-SHOOTER pipeline distribution kit contains:

xshooter-manual-pdf	The X-SHOOTER pipeline manual
install_pipeline	Install script
cpl-6.3.tar.gz	CPL 6.3
esorex-3.10.tar.gz	esorex 3.10
gasgano-2.4.3.tar.gz	GASGANO 2.4.3
xsh-2.2.0.tar.gz	X-SHOOTER 2.2.0
xsh-calib-2.2.0.tar.gz	X-SHOOTER static calibration files 2.2.0

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the X-SHOOTER pipeline 2.2.0 package. It can be any directory of your choice but not:

```
$HOME/gasgano
$HOME/.esorex
```

2. Download from the ESO ftp server, `www.eso.org/pipelines`, the latest release of the X-SHOOTER pipeline distribution.

3. Verify the checksum value of the tar file with the cksum command. **cksum xsh-kit-2.2.0.tar.gz**

4. Unpack using the following commands:
   gunzip xsh-kit-2.2.0.tar.gz tar -xvf xsh-kit-2.2.0.tar

   Note that the size of the installed software (including *Gasgano*) together with the static calibration data is about 27Mb.

5. Set the environment variable JAVA_HOME to the directory where you have the JDK 1.6 or newer installed. If this value is not set, the installation script will try to guess it, but if no JDK is found, the gasgano distribution will not be installed; C-Lapack, CPL, EsoRex and the pipeline will be installed anyway.

6. Install: after moving to the top installation directory,
   cd xsh-kit-2.2.0

   it is possible to perform a simple installation using the available installer script (*recommended*):
   ./install_pipeline

   Note: on recent Mac OS in order to properly install the kit it may be useful to set the following environment variable:

   ```
   export JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/
   ```

7. Check the installation log: probably this will suggest you to set the environment variable CPLDIR and to extend your PATH.

   By default the script will install the X-SHOOTER recipes, *Gasgano*, *EsoRex*, all the necessary libraries, and the static calibration tables, into a directory tree rooted at `$HOME`. A different path may be specified as soon as the script is run.

   The only exception to all this is the *Gasgano* tool, that, if you have the required proper installation of the JDK (version 1.6), will always be installed under the directory `$HOME/gasgano`. Note that the installer will move an existing `$HOME/gasgano` directory to `$HOME/gasgano.old` before the new *Gasgano* version is installed.

   Important: the installation script would ensure that any existing *Gasgano* and *EsoRex* setup would be inherited into the newly installed configuration files (avoiding in this way any conflict with other installed instrument pipelines).