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

Reflex VIMOS/IFU Tutorial

VLT-MAN-ESO-19500-5296

Issue 1.0

Date 2012-06-28

Prepared:	Lodovico Coccato	2012-06-28	
	Name	Date	Signature
Approved:			
	Name	Date	Signature
Released:			
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	15/01/2011	All	Public release

This page was intentionally left blank

Contents

1 Introduction And Scope

`Reflex` is the ESO Recipe Flexible Execution Workbench, an environment to run ESO VLT pipelines which employs a workflow engine (Kepler[1]) to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers. This document is a tutorial designed to enable the user to employ the VIMOS/IFU workflow to reduce his/her data in a user-friendly way, concentrating on high-level issues such as data reduction quality and signal-to-noise (S/N) optimisation.

A workflow accepts science and calibration data, as delivered to PIs in the form of PI-Packs (until October 2011) or downloaded from the archive using the CalSelector tool[2] and organises them into DataSets, where each DataSet contains one science object observation (possibly consisting of several science files) and all associated raw and static calibrations required for a successful data reduction. The data organisation process is fully automatic, which is a major time-saving feature provided by the software. The DataSets selected by the user for reduction are fed through the workflow which executes the relevant pipeline recipes (or stages) in the correct order, providing optional user interactivity at key data reduction points with the aim of enabling the iteration of certain recipes in order to obtain better results. Full control of the various recipe parameters is available within the workflow, and the workflow deals automatically with optional recipe inputs via built-in conditional branches. Additionally, the workflow stores the reduced final data products in a logically organised directory structure and employing user-configurable file names.

IMPORTANT NOTE: The workflow uses the OCA rules defined in the file vimos_mos_wkf.oca and it works for files downloaded from the ESO archive with the CalSelector tool (from year 2009 onwards). For older datasets where the data were directly delivered to the PI (e.g. in DVDs), the file vimos_mos_wkf.dvd.oca should be used instead. Moreover, for dataset in DVD distribution, the user should: i) remove *all* the pipeline products i.e. masterbias (whose filename contains the string `MBIA`), transmission respons files (whose filename contains the string `PTNF`) and directories labeled as `proc` or `reduced`; and ii) remove duplicate files (i.e. files with the same name, but stored in different directories, such as arc lamps).

This tutorial deals with the reduction of VIMOS Integral Field Unit observations only via the VIMOS/IFU workflow. The user is referred to the VIMOS webpage (http://www.eso.org/sci/facilities/paranal/instruments/vimos/) for more information on the instrument itself, and the VIMOS pipeline user manual for the details of the pipeline recipes (http://www.eso.org/sci/software/pipelines/).

[1] https://kepler-project.org
[2] http://www.eso.org/sci/archive/calselectorInfo.html

2 Software Installation

The software pre-requisites for Reflex 2.1 may be found at:

`http://www.eso.org/sci/software/pipelines/reflex_workflows`

To install the Reflex 2.1 software and demo data, please follow these instructions:

1. From any directory, download the installation script:
   `wget ftp://ftp.eso.org/pub/dfs/reflex/install_reflex`
2. Make the installation script executable: `chmod u+x install_reflex`
3. Execute the installation script:
   `./install_reflex`
   and the script will ask you to specify three directories: the download directory `<download_dir>`, the software installation directory `<install_dir>`, and the directory to be used to store the demo data `<data_dir>`. If you do not specify these directories, then the installation script will create them in the current directory with default names.
4. You will be given a choice of pipelines to install. Please specify the numbers for the pipelines you require, separated by a space, or type “A” for all pipelines.
5. To start Reflex, issue the command:
   `<install_dir>/bin/reflex`
   It may also be desirable to set up an alias command for starting the Reflex software, using the shell command alias. Alternatively, the PATH variable can be updated to contain the `<install_dir>/bin` directory

3 Getting started

ESOrex and VIMOS pipeline need to be installed on your machine. You can download them from http://www.eso.o and follow the intallation instructions. We recommend to replace older pipelines version that might be present in your machine with the most updated version, to avoid any compatibility issue.

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 VIMOS/IFU demo data set supplied with the Reflex 2.2 release.

1. Start the Reflex application:

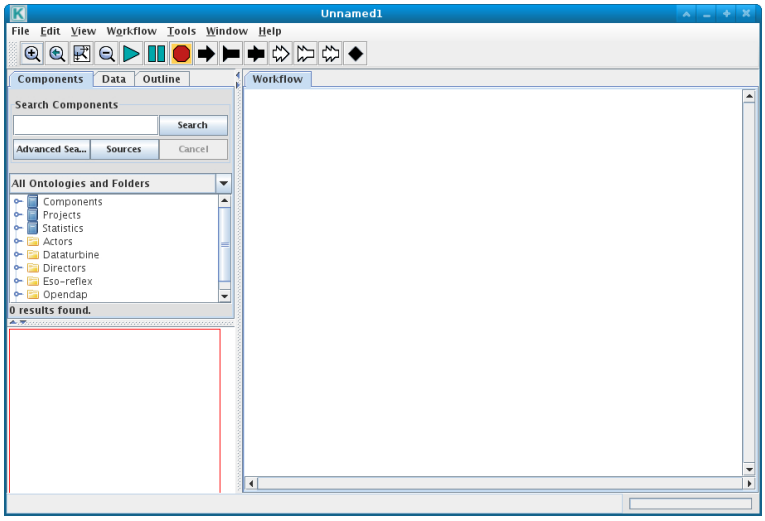

Figure 3.0.0: Empty REFLEX canvas.

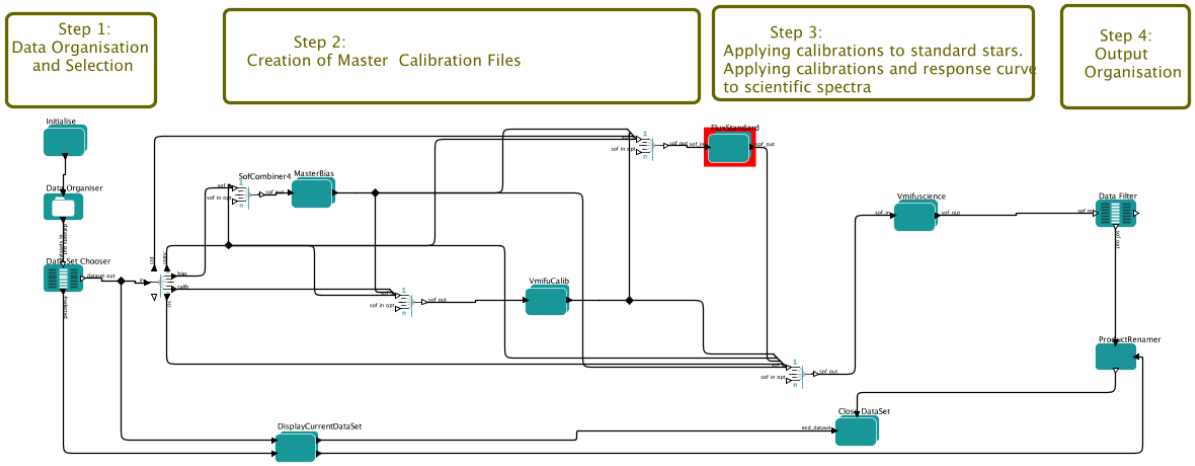

Figure 3.0.0: Zoom on the central part of the REFLEX workflow for reduction of VIMOS/IFU data sets.

```
reflex &
```

The empty `Reflex` canvas as shown in Figure 3.0.0 will appear.

2. Load the VIMOS/IFU workflow Vimos_ifu_v<version_number>.kar from the `File → Open File` menu. The main structure of the VIMOS/IFU workflow is shown in Figure 3.0.0.

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 (10 ms is recommended), and click the "OK" button.

3.1 Workflow configuration

The working directories and the paths of raw science and calibration data, as well as the recipe configuration parameters, need to be specified in the workflow before execution.

3.1.1 Input files

The files location and working directories can be specified directly on the workflow, by double-clicking with the mouse on the relevant paths shown in the upper part of the workflow, and entering the correct path on the window that will pop-up (see Figure 3.1.0.

The parameters to configure are:

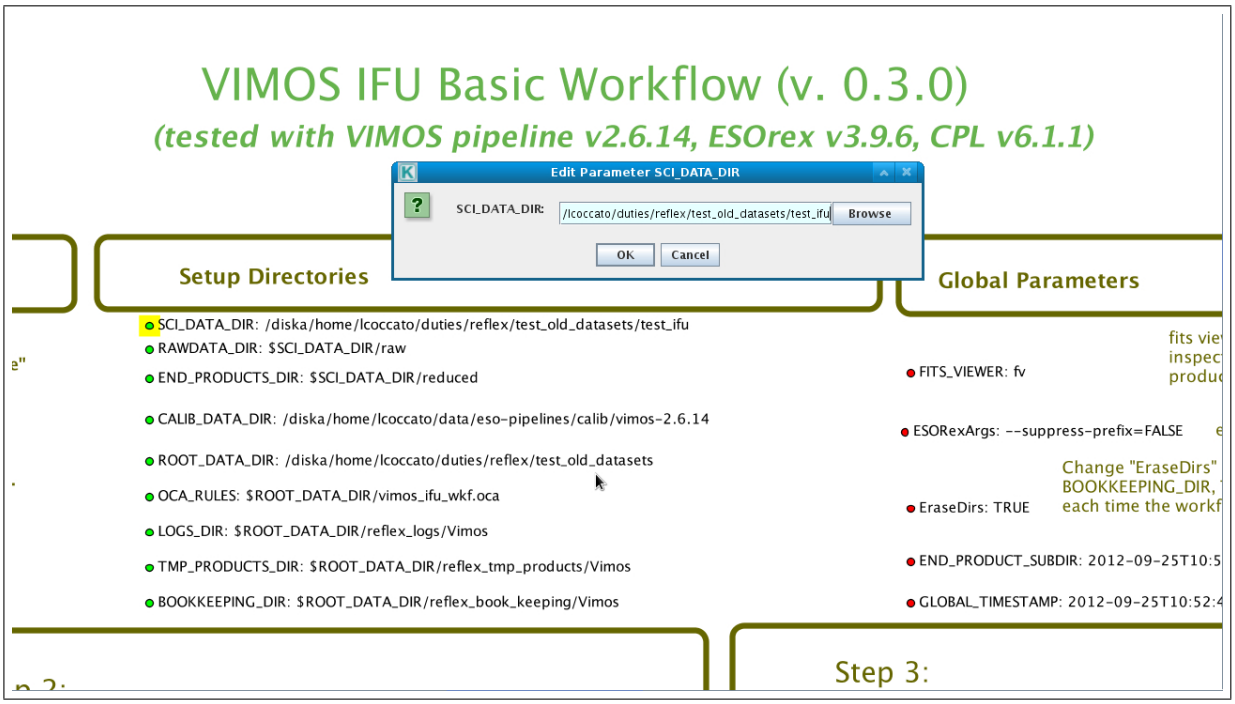

Figure 3.1.0: Zoom on the upper part of the `REFLEX` workflow for reduction of VIMOS/IFU data sets.

- SCI_DATA_DIR. Is specifies the root name of the directories where to find the raw data, and where save the reduced data.
- RAWDATA_DIR. It specifies the directory where the raw data are to be searched for. Sub-directories will be scanned as well.
- END_PRODUCTS_DIR. Is specifies the name of the directory where to save the final products. It must be an existing directory.
- CALIB_DATA_DIR. Directory with static calibration data provided with the VIMOS pipeline distribution. If static calibration data are present in the RAWDATA_DIR (e.g. datasets are downloaded from the archive, or copied from ESO-DVD distribution), then you have to set this directory equal to RAWDATA_DIR (otherwise an obsolete static calibration file may be selected instead of the most appropriate one). Suggestion: if static calibration are missing in the RAWDATA_DIR, it is better if you copy the needed files from the pipeline distribution into RAWDATA_DIR, and let CALIB_DATA_DIR = $RAWDATA_DIR in the configuration parameter.
- OCA_RULES. Name of the files containing the instructions for organization, classification and association of the files. This file is distributed with the workflow.
- ROOT_DATA_DIR. Root name for the directories that will contain the .fits, .log, and .sof files produced by the pipeline.
- TMP_PRODUCTS_DIR. It specifies the directory where the pipeline output fits files will be stored. Final products will be renamed and stored also in END_PRODUCTS_DIR.
- LOGS_DIR. It specifies the directory where the pipeline log files will be stored.
- BOOKKEPING_DIR. It specifies the directory where the .sof files produced by the workflow and used by the pipeline will be stored.

3.1.2 Recipe input configuration parameters

The REFLEX workflows uses the following recipes of the VIMOS/IFU pipeline: vmbias, vmifucalib, vmifustandard, and vmifuscience. The configuration parameters for each recipe can be modified directly on the workflow, by double-clicking with the mouse on the relevant recipe parameter shown in the lower part of the workflow, and entering the correct value on the window that will pop-up (see Figure 3.1.0. The recipe default parameters are listed for reference. We refer the user to the VIMOS pipeline manual for the description of the various recipes and their parameters.

If different datasets need to be reduced with different values for the recipe parameters, it is advisable to use a separate workflow for each dataset.

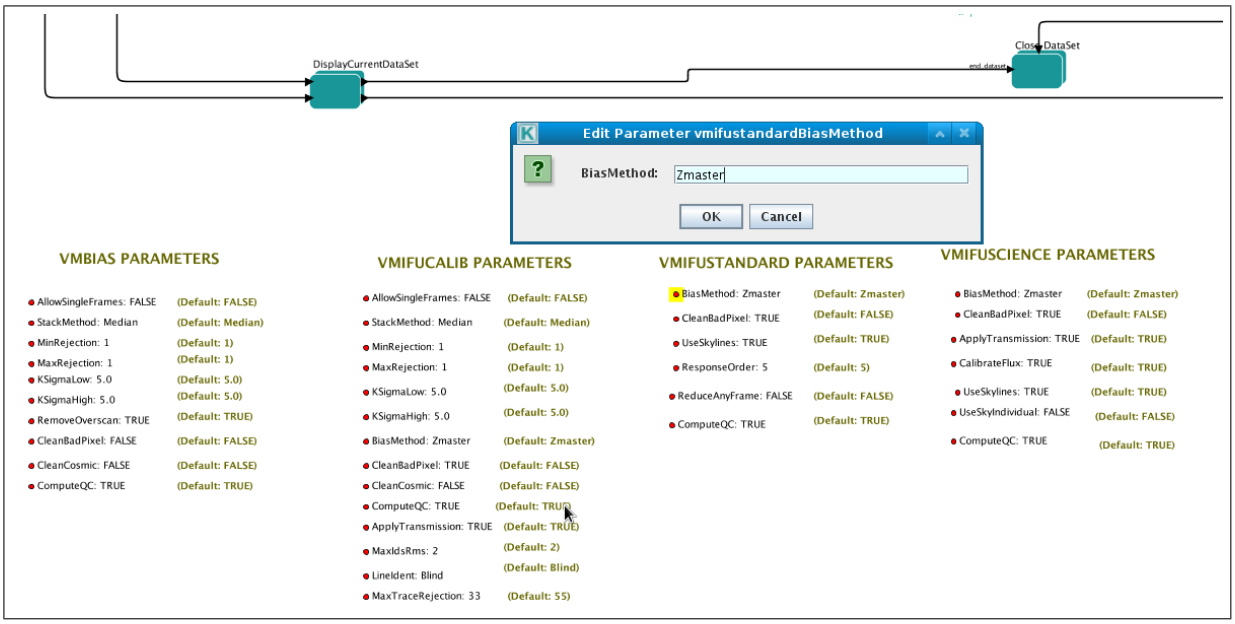

Figure 3.1.0: Zoom on the lower part of the REFLEX workflow for reduction of VIMOS/IFU data sets.

4 Workflow execution

To execute the workflow press the play button on the menu button in the REFLEX main window.

4.1 Initialize

Once started, the workflow will erase the content of the temporary directories REFLEX_BOOK_KEEPING, REFLEX_LOGS, and REFLEX_TMP_PRODUCTS. The content of the reduced data will obviously not be cleaned. To remove this option, just set the parameter EraseDirs to FALSE on the upper right corner of the workflow.

4.2 Data organizer

The workflow will look for the raw data to reduce (calibration, science, and static calibration) in the RAWDATA_DIR. Scientific data are organized in dataset, contining all the calibration, standard stars, and static calibrations associated to that. A dataset can be reduced only if all the needed calibrations are present and properly associated. A list of dataset will be prompted (Figure 4.2.0, and the used can select which dataset to reduce. It is advisable to reduce a dataset per time, and have a separate workflow with optimized recipe parameters for each dataset. If the recipe parameters do not change for different datasets, then the all the datasets can be selected and reduced in the same workflow.

You can see which science and calibration files are associated to which dataset by clicking on the Inspect Highlighed button. The list of files grouped by "Action" (i.e. in which pipelien recipe they are going to be processed) and "Category" (bias, flat field, science, calibration tables, standard stars, etc). Each file can be displayed by the fits viewer specified in the parameter FITS_VIEWER (on the upper right corner of the workflow). The default viewer is fv.

4.3 Recipe execution

Once the datasets to be reduced are selected, press the Continue button on the dataset organized window to proceed with the data reduction. The workflow will automatically execute the pipeline recipes and construct the .sof files to feed the pipeline recipes with. Each sof file will be saved in the BOOKKEPING_DIR directory (and subdirectory within it), depending on the recipe it is associated to and the execution time. The pipeline parameters are specified in the bottm section of the workflow.

4.4 Final products

Once a dataset is reduced (i.e. when the vmifuscience recipe is terminated), a window containing the list of science product is promped. Each file can be inspected with the selected fits viewer. Final science products will be stored in the END_PRODUCTS_DIR, and sorted by excution time and dataset identifier (i.e. the name of the science frame the dataset is for). Default names for the science products are: IFU_<OB name>_IFU_FOV.fits, IFU_<OB name>_IFU_SCIENCE_FLUX_REDUCED.fits, IFU_<OB name>_IFU_SCIENCE_REDUCED.fits, IFU_<OB name>_IFU_SKY_IDS.fits, and IFU_<OB name>_IFU_SKY_TRACE.fits. We refer the user to the VIMOS pipeline manual for the description of these files.

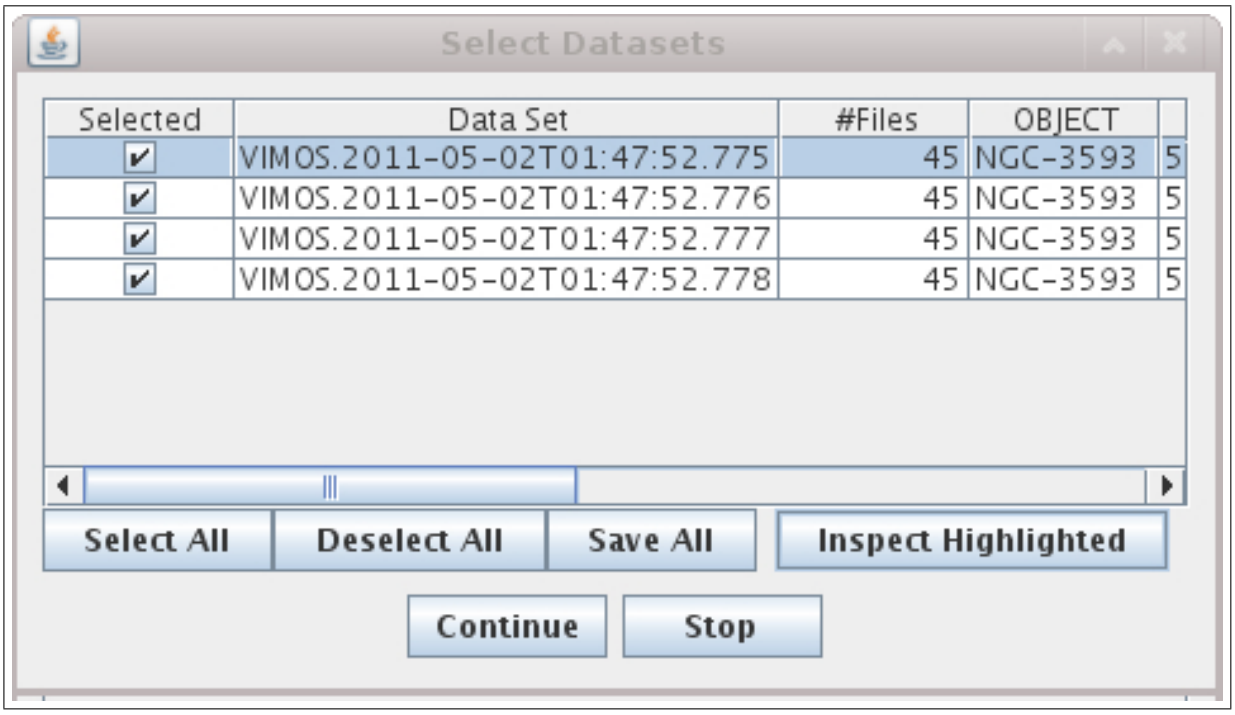

Figure 4.2.0: Data organizer window. Datasets are displayed.

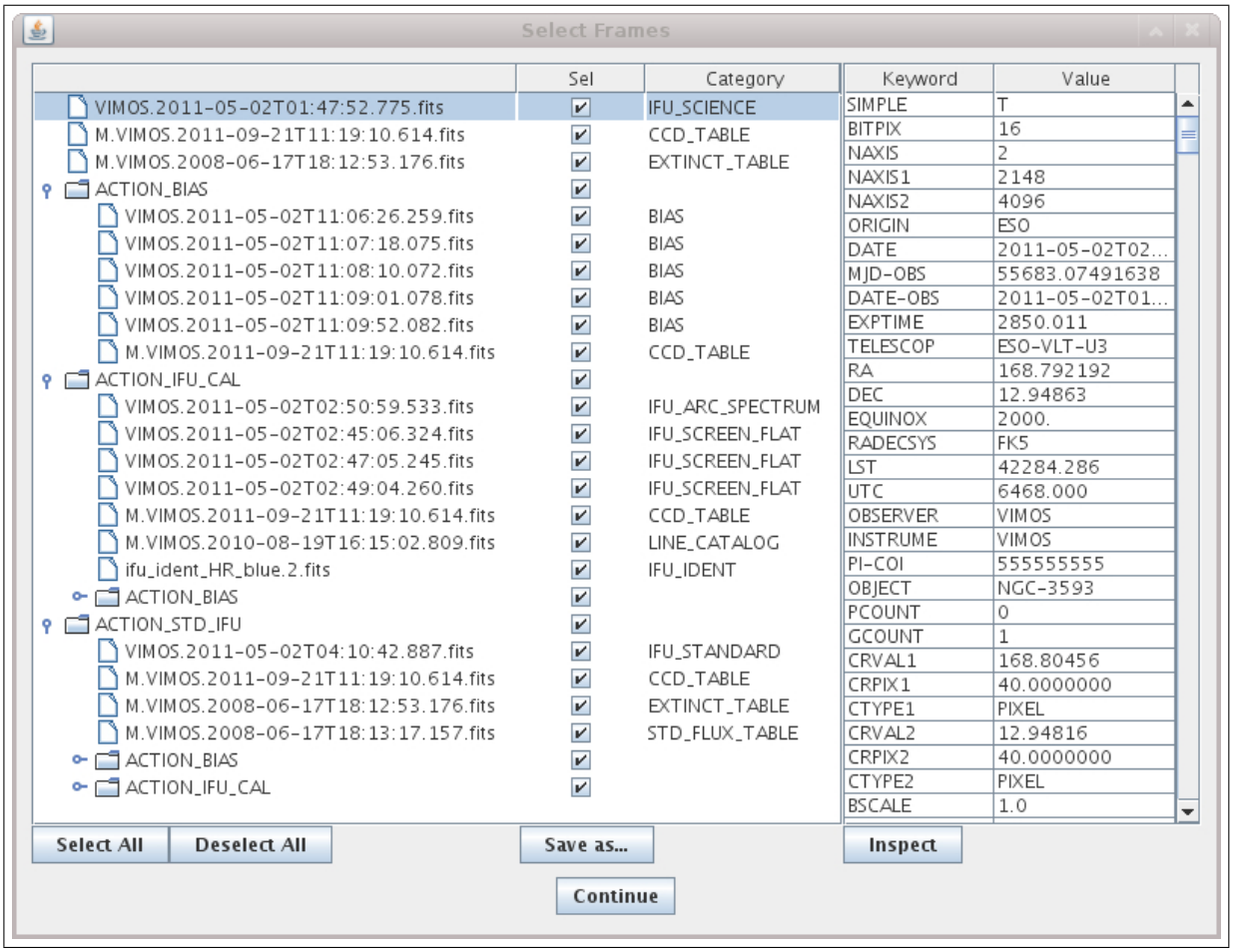

Figure 4.2.0: Data organizer window, showing the files associated to a specific dataset.

5 Troubleshooting

In this section we describe some of the problems that may occur when reducing the VIMOS/IFU with the ESOrex pipeline. For a more comprehensive description we refer the user to the VIMOS user manual (`http://www.eso.org/sci/software/pipelines/`).

5.1 Fiber misidentification

The recipe `vmifucalib` uses the `IFU_IDENT` calibration file [3] to identify the fiber in the flat field. If a fiber identification file is not specified, the fiber spectra identification is still attempted, but the result is not always correct.

The IFU fiber identification performed by recipe `vmifucalib` appears to be negatively affected by changes in temperature. If in the recipe products more than about 50 fibers appear to be "lost" in one pseudo-slit, it may help to rerun the recipe using the `blind` fiber identification method: this method is always triggered if no fiber identification table is specified in the input set-of-frames.

There are 2 main ways to recognize fiber misidentification problems.

1. A defined set of fibers has intentionally null transmission in each quadrant. This help the cross-correlation in the fiber identification process. These dead fibers are 20-21, 60-61, 100-101 and so on, i.e. two fibers each 40. By inspecting the 2D reduced spectrum `SCIENCE_FLUX_REDUCED.fits` it is possible to see whether the “dead fibers” are placed in the correct positions (see Figure 5.1.0).
2. A fiber misidentification would appear later on the reconstructed image of the field-of view (generated by the `vmifuscience` recipe) as zig-zagging patterns breaking the generally smooth look of the intensity distribution. The field of view file `FOV` can be inspected each time the workflow has reduced a dataset setting `mode` to `INSPECT` to the `DataFilter2` actor (right-click with the mouse on the actor, and select `configure actor`).

   An example of the effect that fiber misidentification has on the field-of-view is shown in Figure 5.1.0.

If a fiber misidentification occurs, and the `blind` method in `vmifucal` does not work (see point n° 2 above), one solution could be to manually shuffle the fiber of 1 position, according to the fiber coordinates specified in the `IFU_TABLES` [4]. For example with reference to quadrant number 3, if the fiber n° 290 at pixel coordinates (30,35) on the field of view should be placed in at the pixel coordinates (31,35) instead, because of a fiber mismatch problem, it need to be re-identified as fiber number 291, and so on. The inspection of the re-shuffled FOV helps in understanding if the correction was done in a proper way.

Unfortunately, this shuffle correction is not supported in the VIMOS/IFU reflex workflow, and must be operated by the user on the reduced files on a case-by-case basis.

[3] `IFU_IDENT` consists of intensity profiles (one for each IFU pseudo-slit) cut along the cross-dispersion direction of a reference flat field exposure where the fiber spectra have been safely identified. The fibers corresponding to the peak positions of each profile are listed in the `IFU_IDENT` file. Such safe identifications would then be transferred to the new input flat fields by cross-correlation. In the calibration directories there is ideally one `IFU_IDENT` file for each quadrant/grism combination, named `ifu_ident_grism_q.fits` (where `q` indicates the VIMOS quadrant number, and `grism` the grism name).

[4] The `IFU_TABLES` are static calibration files that contain the correspondence between fiber number in the 2D spectrum and the pixel coordinate on the field of view. See Section 6.21 of the VIMOS manual (version 6.5) for further details.

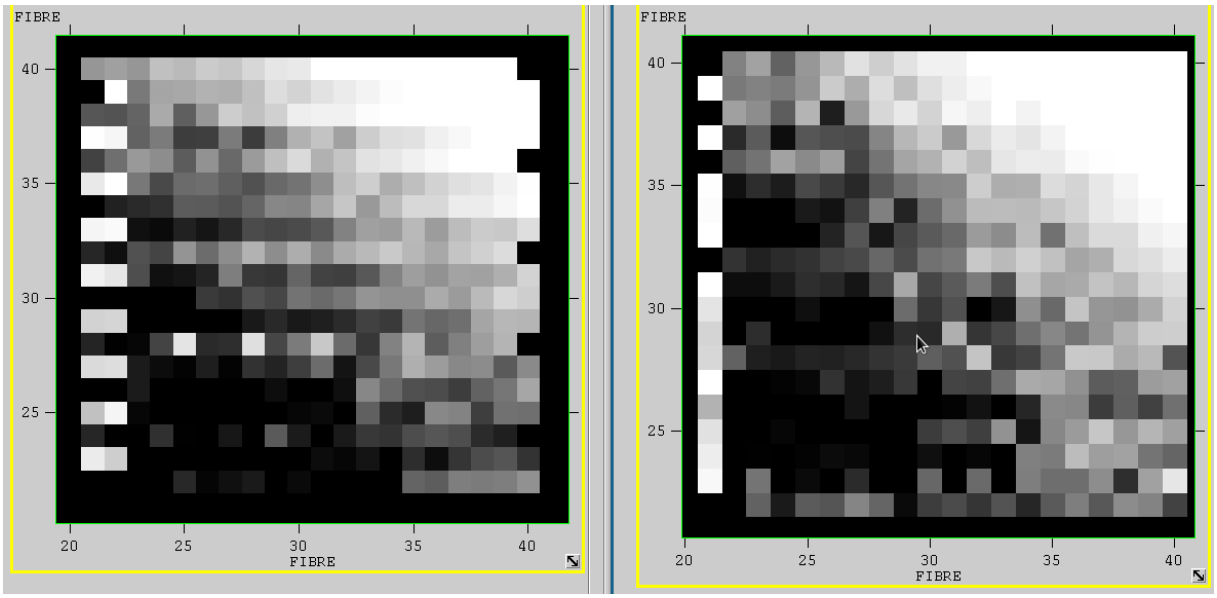

Figure 5.1.0: Example of field of view where fiber misenditification shuffled horizontally the fiber position (left panel), compared to one where fiber identification worked properly (right panel). The field of view refers only to quadrant 3.

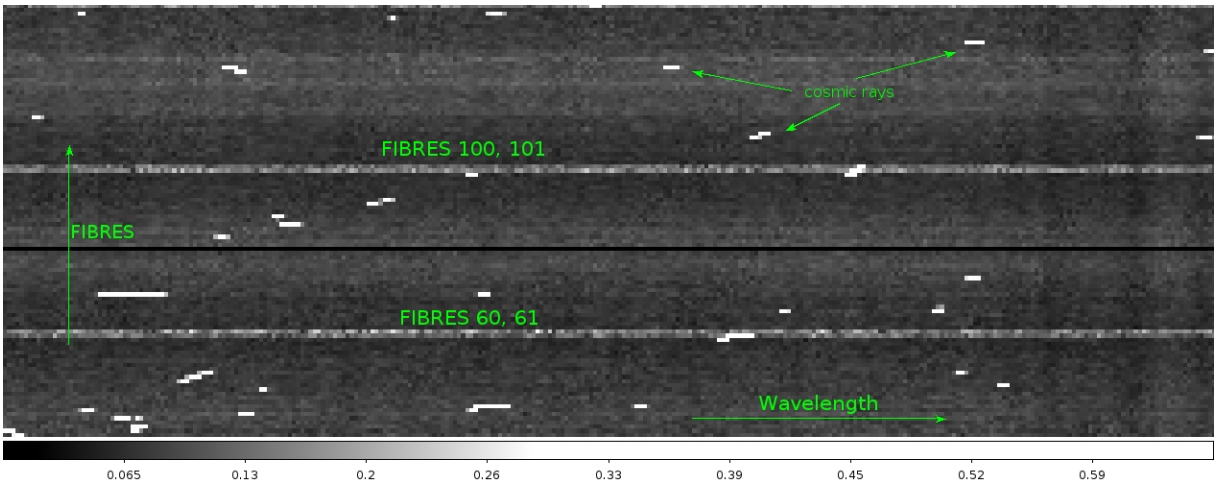

Figure 5.1.0: Example of 2D spectra where the marked fibers (60-61, and 100-101) are properly indentifyed.

5.2 Optimizing the number of recovered fibers.

The fiber identification process operated in `vmifucal` select useful fib-res on the basis of a trace rejection algorithm. The parameter that regulates this procedure is `MaxTraceRejection`. MaxTraceRejection sets the maximum percentage of rejected positions in fiber spectra tracing (default = 50). In the fiber tracing operation, a number of pixel positions may be rejected because the detected position outlays the general trend, or because the signal level is too low. When the percentage of rejected positions is more than what is specified here, then the corresponding fiber is flagged as âĂIJdeadâĂİ and excluded from further processing. This parameter can be modified to recover the majority of useful fibers. For example, for the HR grism the range 10– 80 gives good results. The expected number of good fibres is abput 380 (quadrants 1 and 3), 310 (quadrant 2) and 360 (quadrant 4).

The `MaxTraceRejection` can be optimized in the vimos ifu reflex workflow by modifying the apposite field in the lower part of the reflex canvans (See Fig. 3.1.0).

5.3 Bug: MODE mismatch between BIAS frames and pixel tables

It can happen that the BIAS frames associated to the reduction flow of one IFU observations was taken in the imaging mode. The bias can be used, as there is no difference between bias taken in imaging or IFU modes. Nevertheless, the `vmbias` recipe requires the bias frame and the bad pixel table[5] to have the same observing mode. This bug can be overcome by changing the `OBSMODE` in the bias frames from `IMG` to `IFU`. This bug is in the `vmbias` recipe, independent from the reflex workflow, and it may be solved in future VIMOS pipeline releases.

[5] `CCD_TABLEs` are static calibration files, one per quadrant, one set per observing mode, containing the list of bad pixels. See Section 6.2 in the VIMOS pipeline user manual (version 6.5).