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 KMOS Tutorial

VLT-MAN-ESO-19500-....

Issue 1.6

Date 2016-05-06

Prepared:	M. Neeser, L. Coccato, Y. Jung, W. Freudling	2016-05-06	
	Name	Date	Signature
Approved:	–		
	Name	Date	Signature
Released:	–		
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section affected	Reason/Initiation/Documents/Remarks
0.1	25/06/2013	All	Initial Release
0.2	15/07/2013	All	Description of Demo Data
1.0	19/09/2013	All	Revision of product data description and incl. of interactive mode
1.1	29/10/2013	All	Use Reflex 2.5 and Integration in ESO SVN
1.2	19/03/2014	§9	Description of interactive actors (v. 1.3.0), including parameter tables
1.2.1	28/04/2015	§5	Inclusion of new files in the demo dataset.
1.2.5	29/06/2015	All; incl. new §2	Update of figures and text to sync with v. 1.3.12.
1.3	29/06/2015	All	Update of figures and text to sync with v. 1.3.13 and Reflex 2.8.
1.4	07/10/2015	All	Update of figures and text to sync with v. 1.3.14.
1.5	01/01/2016	2, 4, 5, 7, 10	In sync with version 1.3.16
1.6	22/01/2016	All	In sync with version 1.3.17
1.7	22/01/2016		In sync with version 1.3.18
2.0	11/10/2016	4-6,8-10	In sync with version 1.4.0

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 to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers. The basic philosophy and concepts of Reflex have been discussed by Freudling et al. (2013A&A...559A..96F). Please reference this article if you use Reflex in a scientific publication.

Reflex and the data reduction workflows have been developed by ESO and instrument consortia and they are fully supported. If you have any issue, please contact *usd-help@eso.org* for further support.

This document is a tutorial designed to enable the user to employ the KMOS 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 downloaded from the archive using the CalSelector tool[1] (with associated raw calibrations) 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 to the workflow which executes the relevant pipeline recipes (or stages) in the correct order. 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 employing user-configurable file names.

This tutorial deals with the reduction of KMOS IFU observations only via the KMOS `Reflex` workflow. The user is referred to the KMOS user manual (Cirasuolo 4[2]) and to the ESO instrument web pages [3] for more information on the instrument itself as well as a summary of available documentation, recent news, and tools. A brief description of the KMOS data flow is given in Davies 2[4].

The quick start section (see Section 5) describes the minimum effort to get started, and it makes up only two pages of text in this tutorial. User support for this software is available by sending enquiries to `usd-help@eso.org`.

[1] *http://www.eso.org/sci/archive/calselectorInfo.html*
[2] available at: http://www.eso.org/sci/facilities/paranal/instruments/kmos/doc
[3] http://www.eso.org/sci/facilities/develop/instruments/kmos.html
[4] available at: http://www.mpe.mpg.de/389507/davies-7735-254x.pdf

2 Workflow Status

The KMOS Reflex workflow, in its current version, has reached an advanced level and is capable, together with its underlying KMOS pipeline, of delivering high quality, science-ready data products. The Reflex workflow is built upon the KMOS pipeline delivered by the MPE and improved on by ESO, using experience gained in the first few years of KMOS operations and from the KMOS User community.

The first step of the KMOS Reflex workflow is to organise the data of this complex instrument into an associated, organised, and classified structure based on the calibration type, its proximity in time to the science data, its rotator angle, and its match with the filter and grism. The User will be warned if any calibration frames are missing.

The KMOS Reflex workflow will correct the frames for their dark level and structure, flat-field the data, compute a wavelength solution, apply an illumination correction, apply a standard star flux calibration and telluric correction, and create a cube reconstruction of the science data. A large number of data products are created and retained for the User to assess the quality of the pipeline processing.

During the processing within the Reflex workflow, the User has the ability to modify a large number of pipeline parameters in order to optimise the data processing. This is most conveniently done within the interactive Actors associated with the wavelength calibration, the standard star, and the science data reduction.

Although the KMOS pipeline and Reflex workflow have reached high levels of sophistication, the complexity and variety of KMOS data implies that there will always be room for improvement. During the pipeline development and the experience of KMOS use, the pipeline parameters have been set to default values that deliver the best results for the most cases. However, the User should make an effort to adjust and experiment with the parameters to achieve the best results.

When using the Reflex workflow the User should be aware of a number of issues that affect the current KMOS pipeline. The illumination correction that is currently implemented (i.e. the pipeline recipe *kmos_illumination*) may not always produce the best results. For faint extended sources, this can result in a spatial gradient or the presences of stripes in the field-of-view of the reconstructed image. If this occurs, it may be worth considering not to apply the illumination correction to the reduction of the science frames. Alternatively, you can try another illumination correction that exists within the KMOS pipeline (i.e. *kmo_illumination_flat*). Currently, this recipe has not been implemented in the KMOS Reflex workflow.

It is also worth noting that the background sky subtraction is very sensitive to the quality of the input data and may not always work optimally. This is also true of the telluric correction. Typical for all infrared data, this is very much dependent on the conditions at the time of the observations.

For observations carried on extremely good seeing conditions (e.g., FWHM $\leq$ $0.''4$ arcseconds), point-like sources are undersampled in the chip, therefore the reconstruction algorithm can produce ripples (i.e. a wavy-pattern along wavelength) in the final reconstructed spectrum. This effect can be mitigated, although not fully corrected at the moment, by using the cubic-spline interpolation for both standard star and science observation, and by enlarging the aperture radius over which the standard star spectrum is extracted.

Finally, an optimal source and spectral extraction is not yet available within the KMOS pipeline or workflow. In this case it is up to the User to match a spectral extraction to the characteristics of their sources.

3 Software Installation

Reflex and the workflows can be installed in different ways: via package repositories, via the `install_esoreflex` script or manually installing the software tar files.

The recommended way is to use the package repositories if your operating system is supported. The `macports` repositories support OS X, while the `rpm/yum` repositories support Fedora 20/21/22/23. For any other operating system it is recommended to use the `install_esoreflex` script.

3.1 Installing Reflex workflows via `macports`

This method is supported for the OS X operating system. It is assumed that macports (*http://www.macports.org*) and java are installed. If you have any problem with this installation method, please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/macports.html.

For a quick installation, the following steps will install the ESO pipeline macports repository, the KMOS pipeline, including the Reflex workflow support and Reflex itself:

- Set up the repository:

```
# curl ftp://ftp.eso.org/pub/dfs/pipelines/repositories/macports/setup/Portfile -o Portfile
# sudo port install
# sudo port sync
```

- Install the KMOS pipeline:

```
# sudo port install esopipe-kmos-all
```

3.2 Installing Reflex workflows via `rpm/yum`

This method is supported for Fedora 20/21/22/23 operating systems. If you have any problem with this installation method, please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/rpm.html.

For a quick installation, the following steps will install the ESO pipeline rpm repository, the KMOS pipeline, including the Reflex workflow support and Reflex itself:

- Set up the repository for Fedora 20/21:

```
# sudo yum install yum-utils
# sudo yum-config-manager \
  --add-repo=ftp://ftp.eso.org/pub/dfs/pipelines/repositories/fedora/esorepo.repo
```

- Set up the repository for Fedora 22/23:

```
# sudo dnf install dnf-plugins-core
# sudo dnf config-manager \
  --add-repo=ftp://ftp.eso.org/pub/dfs/pipelines/repositories/fedora/esorepo.repo
```

- Install the KMOS pipeline (Fedora 20/21):
  `# sudo yum install esopipe-kmos-all`
- Install the KMOS pipeline (Fedora 22/23):
  `# sudo dnf install esopipe-kmos-all`

3.3 Installing Reflex workflows via `install_esoreflex`

The software pre-requisites for `Reflex 2.8` may be found at:
http://www.eso.org/sci/software/pipelines/reflex_workflows

To install the `Reflex 2.8` 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_esoreflex
   ```

2. Make the installation script executable:

   ```
   chmod u+x install_esoreflex
   ```

3. Execute the installation script:

   ```
   ./install_esoreflex
   ```

   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 asked whether you want to use your Internet connection. Unless you want to reuse already downloaded packages (only advanced users), use the default Yes.

5. You will be given a choice of pipelines (with the corresponding workflows) to install. Please specify the numbers for the pipelines you require, separated by a space, or type "A" for all pipelines.

6. For the pipelines to be installed you will be prompted for the demo data sets to be installed. Type "A" for all demo datasets. Take into account that if you are installing in a directory that already contains data, it won't be removed.

7. The script will also detect whether previous versions of the workflows or Reflex were installed and in this case you have the option to update links or remove obsolete cache directories. It is advised to use the defaults.

8. To start `Reflex`, issue the command:

```
<install_dir>/bin/esoreflex
```

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.

4 Demo Data

Together with the pipeline you will also recieve two demo data sets, that allow you to run the `Reflex KMOS` workflow without any changes in parameters. This way you have data sets to experiment with before you start to work on your own data.

Note that you will need a minimum of ∼5 GB, ∼9 GB and ∼6.9 GB of free disk space for the directories `<download_dir>`, `<install_dir>` and `<data_dir>`, respectively. Also, you need at least ∼9 Gb for processing the demo datasets once with default parameters.

The raw input consists of:

1. a single SCIENCE OB of the HII, star-formation region Gum 43 (RCW 65) executed in NOD to SKY mode.
2. a single SCIENCE OB of a portion of the Orion Nebulae observed in mosaic mode (mapping8).

Both datasets include the raw calibration frames: darks, lamp flats, and arc lamps. For the first dataset, also telluric standards are available, along with the necessary calibrations. The static calibration files (arc line lists, atmosphere model, OH spectral lines, and arc reference lines) are included in the pipeline distribution. The raw tutorial data sets are summarized in tables 4.1 and 4.2.

Table 4.1: The KMOS Reflex workflow tutorial data set, nod to sky mode.

File	DPR.CATG	DPR.TYPE	INS.GRAT1.ID	DET.SEQ1.DIT
KMOS.2013-06-27T02:30:02.036.fits	CALIB	FLAT,OFF	K	3
KMOS.2013-06-27T02:30:13.935.fits	CALIB	FLAT,OFF	K	3
KMOS.2013-06-27T02:30:26.009.fits	CALIB	FLAT,OFF	K	3
KMOS.2013-06-27T02:33:25.442.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:33:37.272.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:33:49.244.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:34:42.585.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:34:54.500.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:35:06.539.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:35:59.909.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:36:11.744.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:36:23.588.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:37:17.002.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:37:27.878.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:37:40.018.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:38:33.399.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:38:45.321.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:38:57.219.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:39:50.556.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:40:02.535.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:40:14.442.fits	CALIB	FLAT,LAMP	K	3
KMOS.2013-06-27T02:42:34.927.fits	CALIB	WAVE,OFF	K	4
KMOS.2013-06-27T02:45:31.613.fits	CALIB	WAVE,LAMP	K	4
KMOS.2013-06-27T02:46:24.832.fits	CALIB	WAVE,LAMP	K	4
KMOS.2013-06-27T02:47:18.086.fits	CALIB	WAVE,LAMP	K	4
KMOS.2013-06-27T02:48:12.603.fits	CALIB	WAVE,LAMP	K	4
KMOS.2013-06-27T02:49:06.959.fits	CALIB	WAVE,LAMP	K	4
KMOS.2013-06-27T02:50:01.906.fits	CALIB	WAVE,LAMP	K	4
KMOS.2013-06-28T22:21:22.894.fits	CALIB	FLAT,SKY	K	5
KMOS.2013-06-28T22:21:57.131.fits	CALIB	FLAT,SKY	K	30
KMOS.2013-06-28T22:22:36.403.fits	CALIB	FLAT,SKY	K	30
KMOS.2013-06-28T22:23:14.587.fits	CALIB	FLAT,SKY	K	30

Table 4.1 – continued from previous page

File	DPR.CATG	DPR.TYPE	INS.GRAT1.ID	DET.SEQ1.DIT
KMOS.2013-06-30T14:52:42.168.fits	CALIB	DARK	YJ	100
KMOS.2013-06-30T14:54:30.888.fits	CALIB	DARK	YJ	100
KMOS.2013-06-30T14:56:19.580.fits	CALIB	DARK	YJ	100
KMOS.2013-06-30T14:58:08.242.fits	CALIB	DARK	YJ	100
KMOS.2013-06-30T14:59:56.907.fits	CALIB	DARK	YJ	100
KMOS.2013-06-30T23:34:43.453.fits	CALIB	OBJECT,SKY,STD,FLUX	K	20
KMOS.2013-06-30T23:35:22.168.fits	CALIB	OBJECT,SKY,STD,FLUX	K	20
KMOS.2013-06-30T23:36:00.586.fits	CALIB	OBJECT,SKY,STD,FLUX	K	20
KMOS.2013-06-30T23:36:36.792.fits	CALIB	OBJECT,SKY,STD,FLUX	K	20
KMOS.2013-06-30T23:48:06.049.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-06-30T23:53:23.571.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-06-30T23:59:09.586.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:04:22.390.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:09:35.560.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:14:52.379.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:20:10.285.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:25:24.507.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:30:37.274.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:35:55.867.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:41:13.785.fits	SCIENCE	OBJECT,SKY	K	300
KMOS.2013-07-01T00:46:26.588.fits	SCIENCE	OBJECT,SKY	K	300

Table 4.2: The KMOS Reflex workflow tutorial data set, mapping8 mode.

File	DPR.CATG	DPR.TYPE	INS.GRAT1.ID	DET.SEQ1.DIT
KMOS.2014-02-18T18:36:02.524.fits	CALIB	DARK	H	60
KMOS.2014-02-18T18:37:11.214.fits	CALIB	DARK	H	60
KMOS.2014-02-18T18:38:19.910.fits	CALIB	DARK	H	60
KMOS.2014-02-18T23:10:29.633.fits	CALIB	FLAT,SKY	HK	5
KMOS.2014-02-18T23:11:08.159.fits	CALIB	FLAT,SKY	HK	15
KMOS.2014-02-18T23:11:32.440.fits	CALIB	FLAT,SKY	HK	15
KMOS.2014-02-18T23:11:55.627.fits	CALIB	FLAT,SKY	HK	15
KMOS.2014-02-19T09:46:28.621.fits	CALIB	DARK	HK	100
KMOS.2014-02-19T09:48:16.258.fits	CALIB	DARK	HK	100
KMOS.2014-02-19T09:50:05.021.fits	CALIB	DARK	HK	100
KMOS.2014-02-19T09:51:53.801.fits	CALIB	DARK	HK	100
KMOS.2014-02-19T09:53:42.505.fits	CALIB	DARK	HK	100
KMOS.2014-02-19T20:28:45.465.fits	CALIB	FLAT,OFF	HK	3
KMOS.2014-02-19T20:28:57.238.fits	CALIB	FLAT,OFF	HK	3
KMOS.2014-02-19T20:29:09.010.fits	CALIB	FLAT,OFF	HK	3
KMOS.2014-02-19T20:32:04.250.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:32:16.041.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:32:27.819.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:33:20.938.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:33:32.708.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:33:44.495.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:34:37.648.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:34:48.348.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:35:00.112.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:35:53.279.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:36:05.085.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:36:16.853.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:37:09.995.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:37:21.795.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:37:33.565.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:38:26.711.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:38:38.502.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T20:38:50.266.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-19T21:00:19.211.fits	CALIB	WAVE,OFF	HK	5
KMOS.2014-02-19T21:03:24.754.fits	CALIB	WAVE,LAMP	HK	5

Table 4.2 – continued from previous page

File	DPR.CATG	DPR.TYPE	INS.GRAT1.ID	DET.SEQ1.DIT
KMOS.2014-02-19T21:04:19.980.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-19T21:05:14.213.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-19T21:06:09.480.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-19T21:07:04.782.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-19T21:08:00.111.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-21T02:45:45.986.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T02:49:23.303.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T02:53:16.772.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T02:56:52.367.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:00:30.537.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:04:08.708.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:07:44.302.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:11:23.330.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:15:00.649.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:18:35.393.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:22:13.557.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:25:51.360.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:29:26.737.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T03:33:01.297.fits	SCIENCE	OBJECT,SKY	HK	200
KMOS.2014-02-21T10:22:41.499.fits	CALIB	DARK	H	60
KMOS.2014-02-21T10:23:50.175.fits	CALIB	DARK	H	60
KMOS.2014-02-21T10:24:58.863.fits	CALIB	DARK	H	60
KMOS.2014-02-21T10:26:07.541.fits	CALIB	DARK	H	60
KMOS.2014-02-21T10:27:16.213.fits	CALIB	DARK	H	60
KMOS.2014-02-21T10:28:26.748.fits	CALIB	DARK	H	200
KMOS.2014-02-21T10:31:54.359.fits	CALIB	DARK	H	200
KMOS.2014-02-21T10:35:23.101.fits	CALIB	DARK	H	200
KMOS.2014-02-21T10:38:51.844.fits	CALIB	DARK	H	200
KMOS.2014-02-21T10:42:20.728.fits	CALIB	DARK	H	200
KMOS.2014-02-21T10:48:19.917.fits	CALIB	FLAT,OFF	HK	3
KMOS.2014-02-21T10:48:31.686.fits	CALIB	FLAT,OFF	HK	3
KMOS.2014-02-21T10:48:43.441.fits	CALIB	FLAT,OFF	HK	3
KMOS.2014-02-21T10:50:35.352.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:50:47.110.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:50:58.874.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:51:51.911.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:52:03.719.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:52:15.484.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:53:08.575.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:53:19.330.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:53:31.087.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:54:24.198.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:54:35.986.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:54:47.736.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:55:40.867.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:55:52.668.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:56:04.428.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:56:57.576.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:57:09.379.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:57:21.116.fits	CALIB	FLAT,LAMP	HK	3
KMOS.2014-02-21T10:59:35.923.fits	CALIB	WAVE,OFF	HK	5
KMOS.2014-02-21T11:02:41.395.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-21T11:03:36.607.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-21T11:04:31.792.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-21T11:05:27.056.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-21T11:06:21.194.fits	CALIB	WAVE,LAMP	HK	5
KMOS.2014-02-21T11:07:16.488.fits	CALIB	WAVE,LAMP	HK	5

5 Quick Start: Reducing The Demo Data

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 KMOS demo data set supplied with the `Reflex 2.8` release. By following these steps, the user should have enough information to perform a reduction of his/her own data without any further reading:

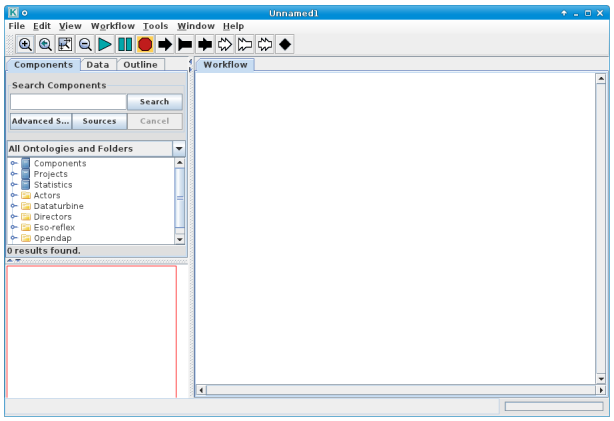


Figure 5.1: *The empty* `Reflex` *canvas.*

1. Start the `Reflex` application:

   ```
   esoreflex &
   ```

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

2. Now open the KMOS workflow by clicking on `File -> Open File`, selecting first `kmos-1.3.20` and then the file `kmos.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 (100 ms is recommended), and click `OK`.

4. Under "Setup Directories" in the workflow canvas there are seven parameters that specify important directories (green dots). Changing the value of `ROOT_DATA_DIR` and/or `RAW_DATA_DIR` is the only necessary modification if you want to process data other than the demo data[5], 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]If you used the install script `install_esoreflex`, then the value of the parameter `ROOT_DATA_DIR` will already be set correctly to the directory where the demo data was downloaded.

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 `RAW_DATA_DIR` under "Setup Directories" in the workflow canvas) and constructs the DataSets. Note that the raw and static calibration data must be present either in `RAW_DATA_DIR` or in `CALIB_DATA_DIR`, otherwise DataSets may be incomplete and cannot be processed. However, if the same reference file was downloaded twice to different places this creates a problem as `Reflex` cannot decide which one to use.
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[6]. The first column consists of a set of tick boxes which allow the user to select the DataSets to be processed. By default all complete DataSets which have not yet been reduced will be selected.
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 reduction of the current DataSet finishes, a pop-up window called *Product Explorer* will appear showing the datasets which have been so far reduced together with the list of final products. This actor allows the user to inspect the final data products, as well as to search and inspect the input data used to create any of the products of the workflow. Figure 5.4 shows the Product Explorer window.
10. The workflow will continue with the remaining DataSets following the same steps described above.
11. After the workflow has finished, all the products from all the DataSets can be found in a directory under `END_PRODUCTS_DIR` with the named with the workflow start timestamp. Further subdirectories will be found with the name of each DataSet.

Well done! You have successfully completed the quick start section and you should be able to use this knowledge to reduce your own data. However, there are many interesting features of `Reflex` and the KMOS workflow that merit a look at the rest of this tutorial.

Note: during the workflow execution, a number of interactive windows associated to specific pipeline recipes will be displayed. In these windows, the user can modify the parameters of corresponding recipe and re-run it. They will be explained in Section 9; press the "Continue" button to proceed with the workflow execution.

5.1 Reducing Your Own Science Data

To reduce your own science data, simply change the path to the raw data directory. This is defined at the top of the workflow window in the area labeled `Setup Directories`. Simply double click on the `RAW_DATA_DIR`, enter the path to your raw science directory and then re-run the workflow in the same way as was done for the tutorial demo data.

The data reduction cascade adopted by the KMOS reflex workflow and the instructions on how to optimize your results will described in Section 9.

[6]The keywords listed can be changed by right-clicking on the `DataOrganiser` Actor, selecting `Configure Actor`, and then changing the list of keywords in the second line of the pop-up window.

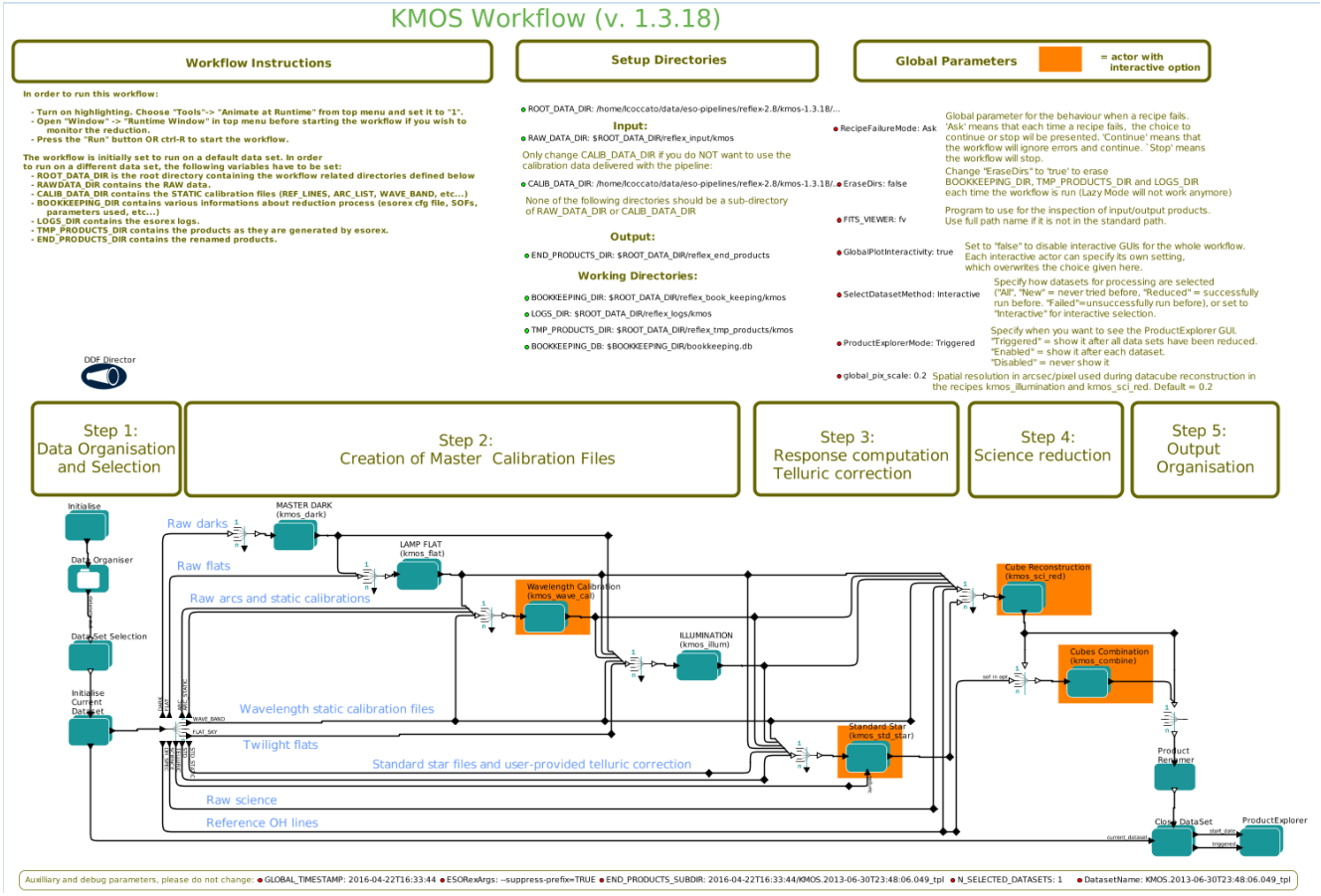


Figure 5.2: *KMOS workflow general layout.*

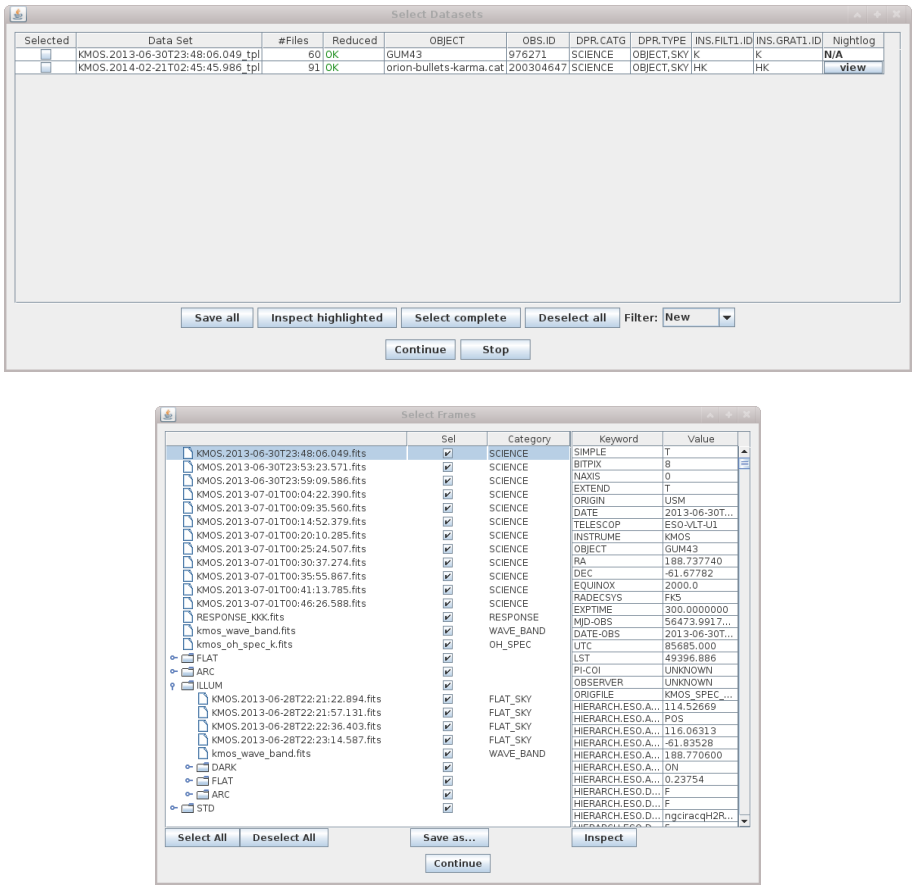


Figure 5.3: Upper panel: the “Select Datasets” pop-up window; Lower panel: the “Selected Frame” pop-up window, obtained after pressing Inspect highlighted in the Select Datasets window.

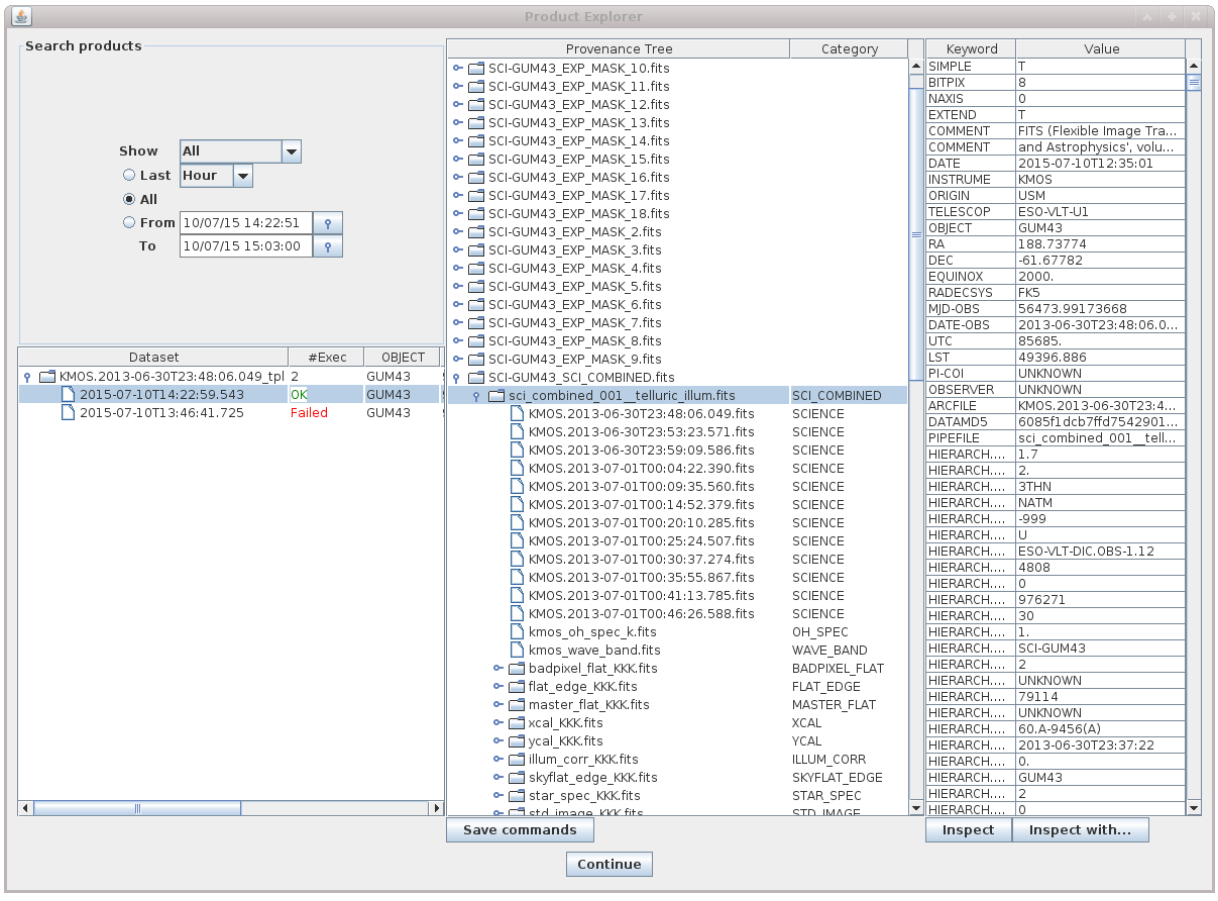


Figure 5.4: *The KMOS product explorer.*

6 KMOS Reduced Data Description

A number of intermediate pipeline products from the tutorial data set can be found in subdirectories of the `TMP_PRODUCT_DIR`. These master calibration files are summarized in Table 6.1.

Table 6.1: The KMOS Reflex workflow tutorial data set: calibration products

File	PRO.CATG	Description
kmos_dark:		
badpixel_dark.fits	BADPIXEL_DARK	preliminary bad pixel map (hot pixels)
master_dark.fits	MASTER_DARK	master dark frame including a noise map
kmos_flat:		
badpixel_flat_KKK.fits	BADPIXEL_FLAT	master bad pixel map (hot + cold pixels)
flat_edge_KKK.fits	FLAT_EDGE	fits table defining the edges of each IFU pseudo slit as derived from the screen flat frames
master_flat_KKK.fits	MASTER_FLAT	master flat-field frame including noise map
xcal_YJYJYJ.fits	XCAL	spatial solution lookup frame
ycal_YJYJYJ.fits	YCAL	spatial solution lookup frame
kmos_wave:		
det_img_wave_KKK.fits	DET_IMG_WAVE	resampled image of reconstructed arc frame
lcal_KKK.fits	LCAL	wavelength solution lookup frame
kmos_illum:		
illum_corr_KKK.fits	ILLUM_CORR	illumination correction to flat-filed
skyflat_edge_KKK.fits	SKYFLAT_EDGE	fits table defining the edges of each pseudo slit IFU as derived from the sky flat frames
kmos_std:		
star_spec_KKK.fits	STAR_SPEC	extracted star spectrum
std_image_KKK.fits	STD_IMAGE	collapsed standard star cube image
std_mask_KKK.fits	STD_MASK	masked pixels within the collapsed standard star cube image
telluric_KKK.fits	TELLURIC	normalized telluric spectrum including noise map

The final products of the reduction pipeline can be found in the directory `END_PRODUCT_DIR` that is defined in the "Setup Directories" section at the top of the workflow.

The science data products from the two tutorial data sets are summarized in tables 6.2 and 6.3. Their description will be done in Section 6.1.

Table 6.2: The KMOS Reflex workflow science products from the tutorial data set nod-to-sky mode (all K-band).

File	Description
PRO.CATG: COMBINE_SCI_RECONSTRUCTED	
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_001.fits	reconstructed science cube for object name '001'
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_002.fits	reconstructed science cube for object name '002'
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_003.fits	reconstructed science cube for object name '003'
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_004.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_007.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_010.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_014.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_018.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_020.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_021.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_027.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_029.fits	"
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_030.fits	"

Table 6.2 – continued from previous page

File	Description
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_048.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_058.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_069.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_100.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_101.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_103.fits	“
PRO.CATG: COMBINE_SCI_RECONSTRUCTED_COLL	
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_001.fits	Image of the field of view for object name ’001’.
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_002.fits	Image of the field of view for object name ’002’
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_003.fits	Image of the field of view for object name ’003’
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_004.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_007.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_010.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_014.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_018.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_020.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_021.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_027.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_029.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_030.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_048.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_058.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_069.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_100.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_101.fits	“
SCI-GUM43_COMBINE_SCI_RECONSTRUCTED_COLL_103.fits	“
PRO.CATG: EXP_MASK_SCI_RECONSTRUCTED	
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_001.fits	Exposure map for object name ’001’
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_002.fits	Exposure map for object name ’002’
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_003.fits	Exposure map for object name ’003’
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_004.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_007.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_010.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_014.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_018.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_020.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_021.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_027.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_029.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_030.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_048.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_058.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_069.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_100.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_101.fits	“
SCI-GUM43_EXP_MASK_SCI_RECONSTRUCTED_103.fits	“
PRO.CATG: SCI_RECONSTRUCTED	
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-06-30T23:48:06.049.fits	Reconstructed cubes for exposure 2013-06-30T23:48:06.049
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-06-30T23:59:09.586.fits	Reconstructed cubes for exposure 2013-06-30T23:59:09.586
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-07-01T00:04:22.390.fits	Reconstructed cubes for exposure 2013-07-01T00:04:22.390
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-07-01T00:09:35.560.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-07-01T00:20:10.285.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-07-01T00:25:24.507.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-07-01T00:30:37.274.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-07-01T00:41:13.785.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-07-01T00:46:26.588.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_KMOS.2013-06-30T23:48:06.049.fits	“
PRO.CATG: SCI_RECONSTRUCTED_COLL	
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-06-30T23:48:06.049.fits	Reconstructed images for exposure 2013-06-30T23:48:06.049

Table 6.2 – continued from previous page

File	Description
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-06-30T23:59:09.586.fits	Reconstructed images for exposure 2013-06-30T23:59:09.586
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-07-01T00:04:22.390.fits	Reconstructed images for exposure 2013-07-01T00:04:22.390
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-07-01T00:09:35.560.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-07-01T00:20:10.285.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-07-01T00:25:24.507.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-07-01T00:30:37.274.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-07-01T00:41:13.785.fits	“
SCI-GUM43_SCI_RECONSTRUCTED_COLL_KMOS.2013-07-01T00:46:26.588.fits	“

Table 6.3: The KMOS Reflex workflow science products from the tutorial data set mapping8 mode (all HK-band).

File	Description
PRO.CATG: COMBINE_SCI_RECONSTRUCTED	
Orion_COMBINE_SCI_RECONSTRUCTED_mapping.fits	Reconstructed cube for the entire mosaic.
PRO.CATG: COMBINE_SCI_RECONSTRUCTED_COLL	
Orion_COMBINE_SCI_RECONSTRUCTED_COLL_mapping.fits	Image of the field of view of the entire mosaic.
PRO.CATG: EXP_MASK_SCI_RECONSTRUCTED	
Orion_EXP_MASK_SCI_RECONSTRUCTED_mapping.fits	Exposure mask of the entire mosaic .
PRO.CATG: SCI_RECONSTRUCTED	
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T02:53:16.772.fits	Reconstructed cubes for exposure 2014-02-21T02:53:16.772
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T02:56:52.367.fits	Reconstructed cubes for exposure 2014-02-21T02:56:52.367
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T03:04:08.708.fits	Reconstructed cubes for exposure 2014-02-21T03:04:08.708
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T03:07:44.302.fits	“
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T03:15:00.649.fits	“
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T03:18:35.393.fits	“
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T03:25:51.360.fits	“
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T03:29:26.737.fits	“
Orion_SCI_RECONSTRUCTED_KMOS.2014-02-21T02:45:45.986.fits	“
PRO.CATG: SCI_RECONSTRUCTED_COLL	
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T02:53:16.772.fits	Reconstructed images for exposure 2014-02-21T02:53:16.772
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T02:56:52.367.fits	Reconstructed images for exposure 2014-02-21T02:56:52.367
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T03:04:08.708.fits	Reconstructed images for exposure 2014-02-21T03:04:08.708
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T03:07:44.302.fits	“
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T03:15:00.649.fits	“
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T03:18:35.393.fits	“
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T03:25:51.360.fits	“
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T03:29:26.737.fits	“
Orion_SCI_RECONSTRUCTED_COLL_KMOS.2014-02-21T02:45:45.986.fits	“

6.1 Description of final products

In this section we provide a short description of the workflow products that are stored in the `reflex_end_products` directory. They are produced by the pipeline recipes *kmos_sci_red* and *kmos_combine*. For further information, please consult the KMOS pipeline manual. In the list, the products are identified by their PRO.CATG keyword.

- `COMBINE_SCI_RECONSTRUCTED`. Product of the recipe *kmos_combine*, which is triggered by the Cubes Combination actor. Its a 3 extensions fits file. The first extension contains the primary header. The second extension contains the fully reduced datacube, obtained by combining the reconstructed cubes

of the IFUs in each exposure that belong to the same object. The datacube is a three-dimensional array (x, y, λ), where the first two dimensions represent the spatial coordinates on the sky (RA and DEC, respectively). The third dimension is wavelength. Therefore for a given (x, y), the datacube shows the spectrum obtained at those RA and DEC coordinates on the sky.

Units are ergs cm^{-2} Å^{-1} s^{-1} for flux calibrated data (as in the nod-to-sky demo dataset) or ADU s^{-1} for non flux calibrated data.

The third extension contains the error cube associated to the first extension. Each dataset produces as many `COMBINE_SCI_RECONSTRUCTED` as targeted objects. In the case of mosaic exposure, only one `COMBINE_SCI_RECONSTRUCTED` is produced.

`COMBINE_SCI_RECONSTRUCTED` are produced by combining the `SCI_RECONSTRUCTED` (products of the recipe *kmos_sci_red*, see below). This is automatically done either i) by combining the extensions of the individual `SCI_RECONSTRUCTED` files that belong to the same target (for nod-to-sky exposures); or ii) by combining all the `SCI_RECONSTRUCTED` for mosaic exposures.

- `COMBINE_SCI_RECONSTRUCTED_COLL`. Product of the recipe *kmos_combine*, which is triggered by the Cubes Combination actor. Reconstructed image associated to the `COMBINE_SCI_RECONSTRUCTED` product. Each dataset produces one `COMBINE_SCI_ RECONSTRUCTED_COLL` for each `COMBINE_SCI_ RECONSTRUCTED`.

  Units are ergs cm^{-2} s^{-1} for flux calibrated data (as in the nod-to-sky demo dataset) or ADU s^{-1} for non flux calibrated data.

- `EXP_MASK_SCI_RECONSTRUCTED`. Product of the recipe *kmos_combine*, which is triggered by the Cubes Combination actor. Exposure map associated to a co-added datacube The value at each pixel tells with how many exposures contributed to that position on the sky.Each dataset produces one `EXP_MASK_ SCI_RECONSTRUCTED` for each `COMBINE_ SCI_RECONSTRUCTED`.

- `SCI_RECONSTRUCTED`. Product of the recipe *kmos_sci_red*, which is triggered by the Cube Reconstruction actor. These files contain, in each extension, a reconstructed datacube associated to a given IFU (which is associated to a target or to a specific location of a mosaic). As in the previous case, each datacube has 3 dimensions (two spatial and one wavelength). The combination of all the reconstructed cubes associated to the same target produces the `COMBINE_SCI_RECONSTRUCTED`. Each dataset produces as many `SCI_RECONSTRUCTED` as exposures in that dataset, that are not dedicated only to sky. Units are ergs cm^{-2} Å^{-1} s^{-1} for flux calibrated data (as in the nod-to-sky demo dataset) or ADU s^{-1} for non flux calibrated data.

- `SCI_RECONSTRUCTED_COLL`. Product of the recipe *kmos_sci_red*, which is triggered by the Cube Reconstruction actor. Reconstructed images associated to the `SCI_RECONSTRUCTED` files. Each extension is associated to a given IFU (which is associated to a target or to a specific location of a mosaic). Each dataset produces as many `SCI_RECONSTRUCTED` as `SCI_RECONSTRUCTED` files.

  Units are ergs cm^{-2} s^{-1} for flux calibrated data (as in the nod-to-sky demo dataset) or ADU s^{-1} for non flux calibrated data.

7 About The Reflex Canvas

7.1 Saving And Loading Workflows

In the course of your data reductions, it is likely that you will customise the workflow for various data sets, even if this simply consists of editing the `ROOT_DATA_DIR` to a different value for each data set. Whenever you modify a workflow in any way, you have the option of saving the modified version to an `XML` file using `File -> Export As` (which will also open a new workflow canvas corresponding to the saved file). The saved workflow may be opened in subsequent `Reflex` sessions using `File -> Open`. Saving the workflow in the default Kepler format (.kar) is only advised if you do not plan to use the workflow with another computer.

7.2 Buttons

At the top of the `Reflex` canvas are a set of buttons which have the following functions:

-  - Zoom in.
-  - Reset the zoom to 100%.
-  - Zoom the workflow to fit the current window size (Recommended).
-  - Zoom out.
-  - Run (or resume) the workflow.
-  - Pause the workflow execution.
-  - Stop the workflow execution.

The remainder of the buttons (not shown here) are not relevant to the workflow execution.

7.3 Workflow States

A workflow may only be in one of three states: executing, paused, or stopped. These states are indicated by the yellow highlighting of the  ,  , and  buttons, respectively. A workflow is executed by clicking the  button. Subsequently the workflow and any running pipeline recipe may be stopped immediately by clicking the  button, or the workflow may be paused by clicking the  button which will allow the current actor/recipe to finish execution before the workflow is actually paused. After pausing, the workflow may be resumed by clicking the  button again.

8 The KMOS Workflow

The KMOS workflow canvas is organised into a number of areas. From top-left to top-right you will find general workflow instructions, directory parameters, and global parameters. In the middle row you will find five boxes describing the workflow general processing steps in order from left to right, and below this the workflow actors themselves are organised following the workflow general steps.

8.1 Workflow Canvas Parameters

The workflow canvas displays a number of parameters that may be set by the user. Under "Setup Directories" the user is only required to set the `RAW_DATA_DIR` to the working directory for the DataSet(s) to be reduced, which, by default, is set to the directory containing the demo data. The `RAW_DATA_DIR` is recursively scanned by the `Data Organiser` actor for input raw data. The directory `CALIB_DATA_DIR`, which is by default within the pipeline installation directory, is also scanned by the `Data Organiser` actor to find any static calibrations that may be missing in your DataSet(s). If required, the user may edit the directories `BOOKKEEPING_DIR`, `LOGS_DIR`, `TMP_PRODUCTS_DIR`, and `END_PRODUCTS_DIR`, which correspond to the directories where book-keeping files, logs, temporary products and end products are stored, respectively (see the Reflex User Manual for further details; [5]).

There is a mode of the `Data Organiser` that skips the built-in data organisation and uses instead the data organisation provided by the CalSelector tool. To use this mode, click on `Use CalSelection associations` in the `Data Organiser` properties and make sure that the input data directory contains the XML file downloaded with the CalSelector archive request.

Under the "Global Parameters" area of the workflow canvas, the user may set the `FITS_VIEWER` parameter to the command used for running his/her favourite application for inspecting FITS files. Currently this is set by default to `fv`, but other applications, such as `ds9`, `skycat` and `gaia` for example, may be useful for inspecting image data. Note that it is recommended to specify the full path to the visualization application (an alias will not work).

By default the `EraseDirs` parameter is set to `false`, which means that no directories are cleaned before executing the workflow, and the recipe actors will work in Lazy mode (see Section 8.2.4), reusing the previous pipeline recipe outputs where input files and parameters are the same as for the previous execution, which saves considerable processing time. Sometimes it is desirable to set the `EraseDirs` parameter to `true`, which forces the workflow to recursively delete the contents of the directories specified by `BOOKKEEPING_DIR`, `LOGS_DIR`, and `TMP_PRODUCTS_DIR`. This is useful for keeping disk space usage to a minimum and will force the workflow to fully rereduce the data each time the workflow is run.

The parameter `RecipeFailureMode` controls the behaviour in case that a recipe fails. If set to `Continue`, the workflow will trigger the next recipes as usual, but without the output of the failing recipe, which in most of the cases will lead to further failrues of other recipes without the user actually being aware of it. This mode might be useful for unattended processing of large number of datasets. If set to `Ask`, a pop-up window will ask whether the workflow should stop or continue. This is the default. Alternatively, the `Stop` mode will stop the workflow execution immediately.

The parameter `GlobalPlotInteractivity` controls whether the interactive windows will appear for those windows which are *enabled* by default. The possible values are `true, false`. Take into account that some

windows are disabled in the default configuration and therefore are not affected by this parameter.

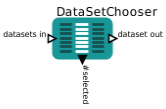
The parameter `ProductExplorerMode` controls whether the `ProductExplorer` actor will show its window or not. The possible values are `Enabled`, `Disabled` and `Triggered`. The latter, recommended, means that the `ProductExplorer` actor will be shown only at the end of the workflow execution.

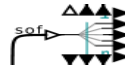
8.2 Workflow Actors

8.2.1 Simple Actors

Simple actors have workflow symbols that consist of a single (rather than multiple) green-blue rectangle. They may also have an icon within the rectangle to aid in their identification. The following actors are simple actors:

- 

- The `Data Organiser` actor.
- 

- The `Data Set Chooser` actor (inside a composite actor).
- 

- The `Fits Router` actor
- 

- The `Product Renamer` actor.
- 

- The `Product Explorer` actor (inside a composite actor).

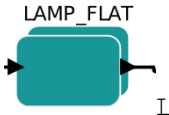
Access to the parameters for a simple actor is achieved by right-clicking on the actor and selecting `Configure Actor`. This will open an "Edit parameters" window. Note that the `Product Renamer` actor is a jython script (Java implementation of the Python interpreter) meant to be customised by the user (by double-clicking on it).

8.2.2 KMOS-specific actors: the workflow data-reduction cascade

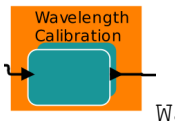
The present KMOS workflow is designed to process the datasets according to a specific data reduction cascade. This cascade triggers a series of pipeline recipes, which are associated to the following composite actors:

- 

`MASTER_DARK`: it executes the recipe *kmos_dark* . It processes the dark frames and creates a master dark frame and bad pixel mask.

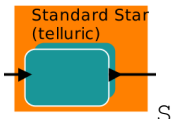


- `LAMP_FLAT`: it executes the recipe *kmos_flat*. It processes the lamp flats. It creates a master flat frame and calibration frames needed for spatial calibration for all three detectors. It also updates the bad pixel mask. It requires the products of *kmos_dark* as input.



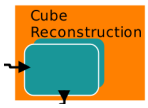
- `Wavelength Calibration`: it executes the recipe *kmos_wave_cal*. It processes the arc frames and creates the wavelength calibration frame needed for all three detectors. It requires the products of *kmos_dark* and *kmos_flat* as input.

- `Illumination`: it executes the recipe *kmos_illumination*. It processes twiligth sky spectra and produces a frame to correct spatial non-uniformity of the flatfield. It requires the products of *kmos_dark*, *kmos_flat*, and *kmos_wave_cal* as input. It is an optional recipe, and it is triggered only if twilight spectra are available in the dataset.

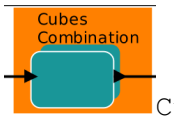


- `Standard Star`: it executes the recipe *kmos_std_star*. It processes standard star observations and generates a response plus telluric correction curve (category `TELLURIC`) . It requires the products of *kmos_dark*, *kmos_flat*, *kmos_wave_cal*, and *kmos_illumination* (if present) as input. It is possible to provide a file with category `TELLURIC` along with the raw data. If the external file is present in the dataset, it will be used to process the science data instead of the one produced from the raw standard stars. The user-provided file must have the same structure than the telluric.fits product of *kmos_std_star*. If the static calibration file containing the response curve (category: `RESPONSE`) is provided, the response corrected 1D extracted stellar spectra are also produced.

  It is an optional recipe, and it is triggered only if standard/telluric stars observations are available in the dataset.



- `Cube Reconstruction`: it executes the recipe *kmos_sci_red*. It processes the raw science frames and create a reduced and sky-subtracted datacube per IFU. It requires the products of *kmos_dark*, *kmos_flat*, *kmos_wave_cal*, *kmos_illumination* (if present), and *kmos_std_star* (if present) as input. The products are or are not flux calibrated, corrected for instrument response and/or telluric absorption depending on the content of the dataset and the value of the **telluric and response correction** workflow parameter (see Section 8.2.3).



- `Cubes combination`: it executes the recipe *kmos_combine*. It combines datacubes belonging to the same target (or the same IFU) into a final datacube (one datacube per object, or IFU). It

requires the products of *kmos_sci_red* as input.

We refer the user to the KMOS pipeline manual for a complete description of the recipes and their parameters.

Some of the above actors trigger an interactive window, allowing the used to inspected the products and, eventually, to re-run the associated pipeline recipe with modified parameters. They are identified by the orange box and will be discussed in details in Section 9.

As noted in the Quick Start Section 5, the workflow will then proceed through its remaining processing steps and write out all pipeline products to the end products directory (specified by the parameter `END_PRODUCTS_DIR` under "Setup Directories" in the workflow canvas). The science data products from the tutorial data set are summarized in section 6 in table 6.2. The intermediate pipeline calibration products can be found in subdirectories of the `TMP_PRODUCT_DIR` and are summarized in section 6 and in table 6.1.

8.2.3 KMOS-specific parameters

The KMOS workflow contains two parameters that are specific to the KMOS data reduction. They are:

- **global_pix_scale**. It specifies the spatial pixel scale in "/pixel of the outputs. It is used by the `ILLUMINATION` and `Cube Reconstruction` actors, which are associated to the recipes *kmos_illum*, and *kmos_sci_red*, respectively. Default=0.2.
- **telluric and response correction** It specifies the strategy of the telluric and instrument response corrections, and the flux calibration to adopt. by selecting which input file has to be used by `Cube Reconstruction`. It can have one of the following values:

  0 This is the default option. Response, telluric correction, and zeropoint are evaluated from standard star observations in the `Standard Star` actor These information are stored into a file with category `TELLURIC`). Alternatively, the user can provide its own `TELLURIC` file into the raw data directory; the user provide file has priority over the one produced by the pipeline. If no `TELLURIC` file is present, the average response curve and zeropoint from static calibration (category `RESPONSE`) are applied if present, otherwise products are not flux calibrated.

  1 Apply only telluric correction and zeropoint from user-provided file with category `TELLURIC_ONLY` in the raw data directory. This file is supposed to contain information only on the telluric correction and zeropoint.

  2 Apply only response correction from static calibration (category: `RESPONSE`). Zeropoint is computed either from standard star or from user-provided file (category: `TELLURIC`). If a `TELLURIC` file is not present, the mean zeropoint information stored in the header of the static calibration file is used. This option is useful, for example when one wants to compute the telluric correction with external tools (e.g. molecfit) and wants the products to be already flux calibrated and response corrected.

  3 Combine response correction from static calibration (category: `RESPONSE`) and telluric correction from user provided file (category: `TELLURIC_ONLY`) and apply it to the science data. The combination of the two corrections are saved into a fits file (category `TELLURIC`). The zeropoint from user provided file is used.

In the following, we describe the content of the TELLURIC, TELLURIC_ONLY, and RESPONSE files:

- TELLURIC. The file contains the combined information of instrument response and telluric correction, and the zeropoint. By defaults, it is created within the Standard Star actor (recipe *kmos_std_star*) by comparing the extracted 1D spectrum of the standard star with a model of that star that takes into account its magnitude and spectral type, and the average grism spectral resolution.
- TELLURIC_ONLY. The file contains only the information of the telluric correction and zeropoint. It has to be prepared by the user using, for example, the molecfit software.
- RESPONSE. The file contains only the median instrument response and the median zeropoint, computed a set of A0 standard stars observed during the instrument lifetime. The responses was extracted by removing the atmospheric telluric contribution (evaluated using molecfit) from the TELLURIC files produced by the pipeline. The median response has errors of $\sim 5\%$ (in the regions not affected by telluric absorptions) up to $\sim 15\%$ (in the regions most affected by telluric absorptions).

8.2.4 Lazy Mode

By default, all recipe executer actors in a pipeline workflow are "Lazy Mode" enabled. This means that when the workflow attempts to execute such an actor, the actor will check whether the relevant pipeline recipe has already been executed with the same input files and with the same recipe parameters. If this is the case, then the actor will not execute the pipeline recipe, and instead it will simply broadcast the previously generated products to the output port. The purpose of the Lazy mode is therefore to minimise any reprocessing of data by avoiding data rereduction where it is not necessary.

One should note that the actor's Lazy mode depends on the contents of the directory specified by BOOKKEEPING_DIR and the relevant FITS file checksums. Any modification to the directory contents and/or the file checksums will cause the corresponding actor when executed to run the pipeline recipe again, thereby rereducing the input data.

The forced rereduction of data at each execution may sometimes be desirable. To force a rereduction of all data for all RecipeExecuter actors in the workflow (i.e. to disable Lazy mode for the whole workflow), set the EraseDirs parameter under the "Global Parameters" area of the workflow canvas to true. This will then remove all previous results as well. To force a rereduction of data for any single RecipeExecuter actor in the workflow (which will be inside the relevant composite actor), right-click the RecipeExecuter actor, select Configure Actor, and uncheck the Lazy mode parameter tick-box in the "Edit parameters" window that is displayed. If the Lazy mode is switched off for an actor, all subsequent actors that use products from that one will also reprocess the data, as they see new products.

9 Optimising Your Results Through Workflow Interaction

In this Section, we use the information from Section 5 along with one KMOS demo data supplied with `Reflex2.8` to illustrate how to optimise the scientific products in terms of quality and signal-to-noise ratio.

The examples shown in this Section refer to the first demo-dataset (object name: Gum 43); informations related to the second dataset will be added whenever relevant.

This is work in progress and the contents of this section will grow as we gain experience and add functionality. Optimising the results is achieved by interaction with the workflow actors via interactive windows displayed at key data reduction points in the data flow, which enable iteration of certain recipes in order to obtain better results.

We recommend that the user has already carried out the reductions for all demo DataSets as described in Section 5, although this is not a pre-requisite to following this section. By doing this, the user will be taking advantage of the workflow Lazy Mode, with minimal waiting time between various pipeline recipe executions.

To start the process, execute the first 5 steps indicated in the Quick Start Section 5.

As discussed previously, the first operation the KMOS workflow does is to group the raw data (science, raw calibrations, and static calibrations) into datasets to be reduced. The datasets will be then processed in series.

The second operation is to direct all the files of a dataset to the correct pipeline recipe, so that the data reduction could start and the various recipes can be triggered in the correct order with the correct set of input files. The data reduction cascade operated by the KMOS workflow triggers the recipes according to the data reduction cascade outlined in Section 8.2.2.

Some of the recipes executed by the workflow trigger an interactive window, allowing the used to inspected the products and, eventually, to re-run the recipe with modified parameters. Section 9.1 describes the common properties of the interactive windows. Sections (9.2 – 9.5) deal with the individual interactive windows.

9.1 General characteristics of the interactive windows

For each of the three interactive windows described above, the panel of buttons at the very top-left of the window may be used to manipulate the displayed plots. The buttons have the following functions:

-  - Reset all the plot ranges to their original values.
-  - Undo the last modification of the plot ranges (if possible).
-  - Redo the next modification of the plot ranges (if possible).
-  - Selecting this button allows the user to use the mouse to shift the plot ranges by left-clicking on the target plot canvas and then dragging the mouse around while keeping the left mouse button held down, and releasing when ready.
-  - Selecting this button allows the user to zoom in on each plot by left-clicking on the target plot canvas to mark the top-left corner of a rectangle and then dragging the mouse to the bottom-right corner of the

rectangle and releasing. The plot ranges will then be modified to match the rectangle that was defined. The following constraints can be made by holding simultaneously a key while pressing the left mouse button:

Constrain pan/zoom to x axis	hold X
Constrain pan/zoom to y axis	hold Y
Preserve aspect ratio	hold Ctrl

-  - Clicking this button opens a "Configure subplots" window that allows the user to adjust the spacing and positioning of the individual plots.
-  - Clicking this button opens a "Save to file" window which allows the user to save a screenshot of the current interactive window.
-  - Clicking this button allows the user to change the display levels of the 2-dimensional merged image by clicking on pixels within the 2-dimensional image (similar to `ds9`).

Use these buttons to inspect the plots in the interactive window in more detail.

Once the user is satisfied with the results, then clicking on the `Continue Wkf` button will continue with the rest of the workflow.

9.2 Wavelength Calibration

In the KMOS workflow, the interactive actor `Wavelength Calibration` is identifiable by an orange rectangle encompassing the actor name. The interactive mode is enabled by default. Should you wish to change that use `Open Actor` to get access to the the components of the interactive actors, then double-click on the composite actors, setting the "EnableInteractivity" parameter to `false`, and clicking `Commit` to save the changes to the workflow.

Figure 9.1 shows the interactive window that will pop-up at the end of the execution of the *kmos_wave_cal* pipeline routine. The image panels at the top of the window show the reconstructed arc frames for each of the three detectors at one of the six rotator position angles. To view the reconstructed arc frames at another rotator angle (either 0, 60, 120, 180, 240, or 300 degrees), simply click on the ovals in the *Angle selection* box. For each of the 18 reconstructed arc frames, the relative average offset of the reconstructed arc lines are plotted in units of pixels for each of the Argon and Neon arc lines (bottom two plots). One-half of the length of the error bars associated with each data point gives the average FWHM (in pixels) for each arc line.

On the right-hand edge of the interactive window the user may modify a number of parameters to improve the wavelength calibration. The parameters and their description is given in table 9.1. If parameters have been changed, then clicking on the `Re-run Recipe` will re-execute the *kmos_wave_cal* pipeline routine.

Other recipe parameters, not present in Table 9.1, can be modified by opening the `Wavelength Calibration` actor, double clicking on the RecipeExecuter associated to the *kmos_wave_cal* recipe (the green box named `kmos_wave_cal_1`), and editing the desired parameter in the apposite field.

You can re-use your preferred parameter set as initial values for subsequent recipe executions by clicking the corresponding button. The new values will be stored for this dataset, but they will be lost afterwards. To

save user-customized values as default open the `Wavelength Calibration` composite actor, identify the parameter you want to change the default from the list (the `init_` parameters identified with the red bullet in the sub-workflow canvas), double-click to edit. These values will be used till the workflow is open, but they will be lost when closing it, unless you save it.

The bottom right list indicates all the files that were used as input to the interactive window, identified by their category. If the user wants to inspect a particular file or assess the recipe products with another viewer or script, (s)he can copy the full file path from the list.

It could happen that the Argon lines have higher absolute residuals than Neon lines, as shown in figure 9.1 for the first demo dataset. This might be due to the fact that Ne lines are more numerous and on average more intense than Argon lines, therefore they drive the wavelength calibration. The difference is less than a pixel and negligible if compared to uncertainties. However, the effect can be mitigated by increasing the `-order` parameter to 7 and press the `Re-run Recipe` button.

Note: `-order=0` uses the grism hard-coded value, which is 6 for the H, K, and YJ grisms, 4 for IZ grism, and 5 for HK grism.

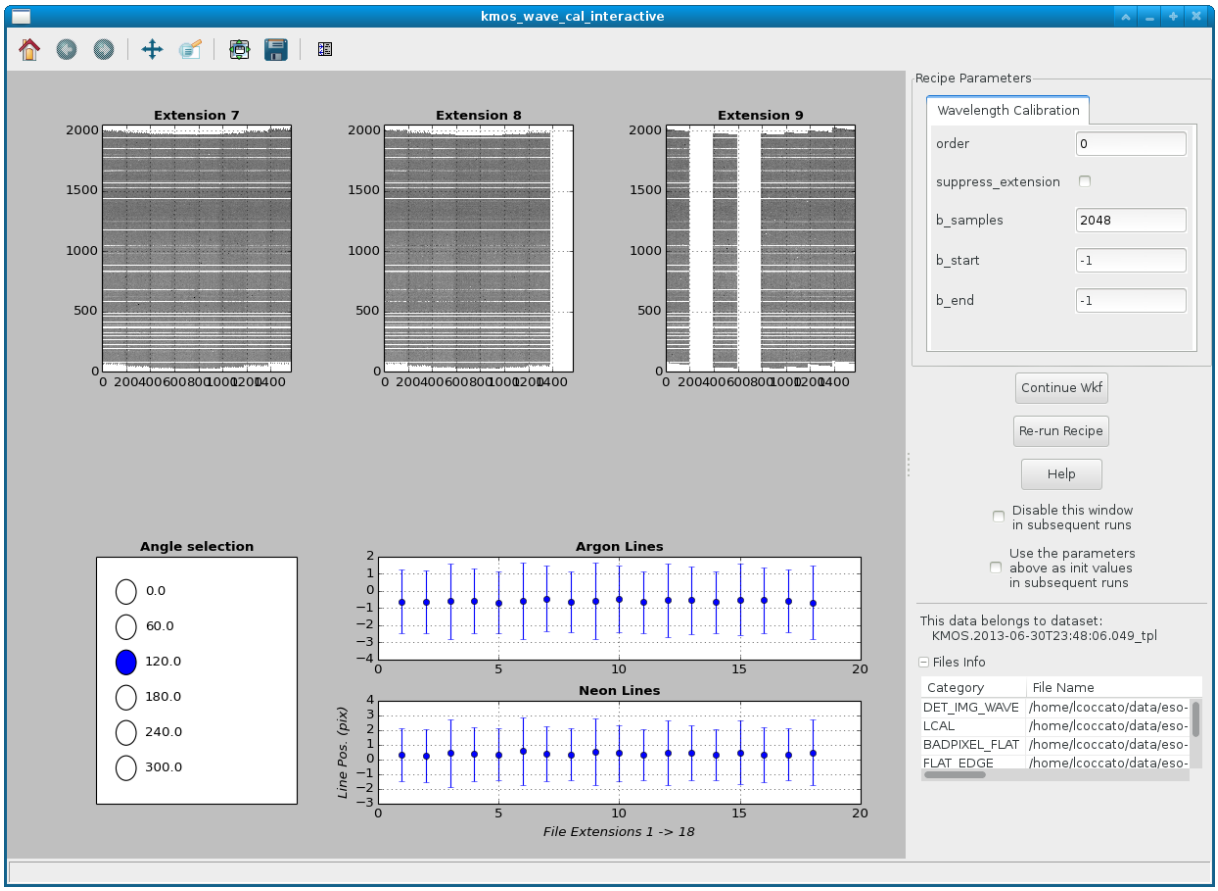


Figure 9.1: *The interactive window for the* `Wavelength Calibration` *actor as displayed for the KMOS tutorial data set..*

Table 9.1: Parameters that a user can manipulate within the WAVE_CAL interactive window

Parameter[1]	Value (default)	Explanation
–order	0	The polynomial order to use for the fit of the wavelength solution. The appropriate order is chosen automatically depending on the waveband. Otherwise an order of 6 is recommended, except for IZ-band, there order 4 should be used
–suppress_extension	FALSE	Suppress arbitrary filename extension. (if TRUE (apply) or FALSE (do not apply)
–b_samples	2048	The number of samples in wavelength for the reconstructed cube
–b_start	-1	The lowest wavelength [μm] to take into account when reconstructing (default of -1 sets the proper value for the actual band automatically)
–b_end	-1	The highest wavelength [μm] to take into account when reconstructing (default of -1 sets the proper value for the actual band automatically)

[1] All parameters, available to the *kmos_wave_cal* pipeline routine, have been included for use in the WAVE_CAL interactive window.

9.3 Telluric Standard Star Calibration

In the KMOS workflow, the interactive actor Standard Star is identifiable by an orange rectangle encompassing the actor name. The interactive mode is enabled by default. Should you wish to change that use Open Actor to get access to the the components of the interactive actors, then double-click on the composite actors, setting the "EnableInteractivity" parameter to false, and clicking Commit to save the changes to the workflow.

Figure 9.2 shows the interactive window that will pop-up at the end of the execution of the *kmos_std_star* pipeline routine. Note that for the orion nebulae dataset, there are no standard stars, therefore the recipe and the interactive window are not triggered. The image panel at the top-right of the window shows the median collapsed data cube (pipeline product: STD_IMAGE) for a single IFU. The total number of IFU's dedicated to observing a telluric standard star are listed to the left of this image and clicking on the circle will display the standard star from another IFU. Typically, 3 standard stars are observed (one per detector). If less than one star per detector is observed, the final products will not be flux calibrated nor telluric corrected.

At the bottom of the interactive window the extracted standard star spectrum (for the selected IFU) is displayed (pipeline product: STAR_SPEC) in flux units of ADU and wavelength units of microns.

On the right-hand edge of the interactive window the user may modify a number of parameters to improve the telluric standard star calibration. The parameters and their description is given in table 9.2. If parameters have been changed, then clicking on the Re-run Recipe will re-execute the *kmos_std_star* pipeline routine.

You can re-use your preferred parameter set as initial values for subsequent recipe executions by clicking the corresponding button. The new values will be stored for this dataset, but they will be lost afterwards. To save user-customized values as default open the Standard Star composite actor, identify the parameter you want to change the default from the list (the init_ parameters identified with the red bullet in the sub-workflow canvas), double-click to edit. These values will be used till the workflow is open, but they will be lost when closing it, unless you save it.

The bottom right list indicates all the files that were used as input to the interactive window, identified by their category. If the user wants to inspect a particular file or assess the recipe products with another viewer or script, (s)he can copy the full file path from the list.

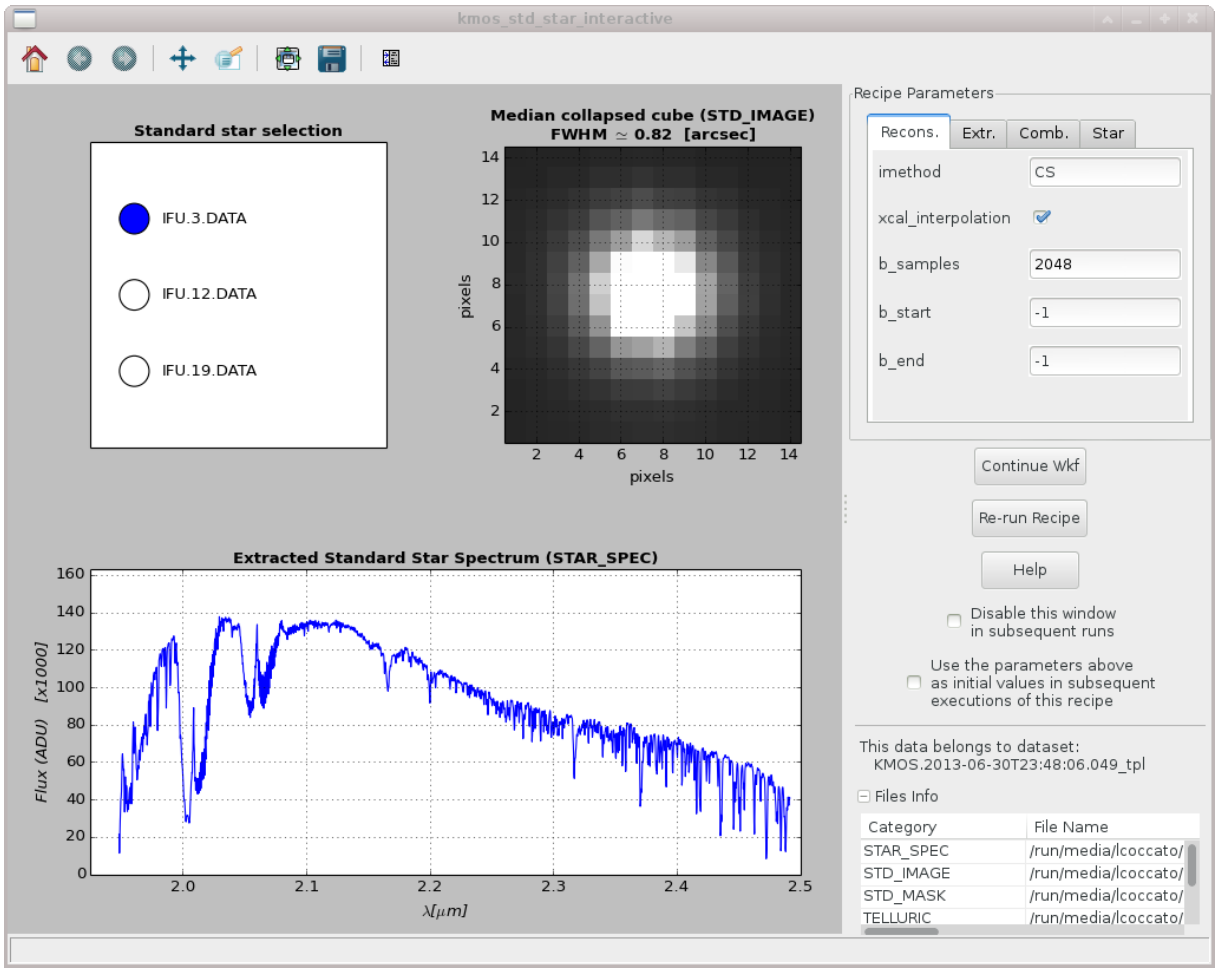


Figure 9.2: *The interactive window for the* `Telluric Standard Star Calibration` *actor as displayed for the KMOS tutorial data set.*.

9.4 Reduction of science frames: reconstruction of datacubes

In the KMOS workflow, the interactive actor `Cube Reconstruction` is identifiable by an orange rectangle encompassing the actor name. The interactive mode is enabled by default. Should you wish to change that use `Open Actor` to get access to the the components of the interactive actors, then double-click on the composite actors, setting the "EnableInteractivity" parameter to `false`, and clicking `Commit` to save the changes to the workflow.

Figure 9.3 shows the interactive window that will pop-up at the end of the execution of the *kmos_sci_red* pipeline routine.

The selection panel at the top of the window shows the list of individual reconstructed science images (pipeline products: `SCI_RECONSTRUCTED`) in the processed data set. Each file can be selected by clicking on the oval with the left mouse button.

The central selection panel summarizes the status of each individual IFU for the selected frame. IFU's for which an extracted spectrum can be viewed are shown as `Active` And are plotted in green. The other IFU statuses, in `Collision`, `NotInPAF`, `Empty`, `Locked` and `NotInPAF&Locked` are shown in cyan, blue, black, magenta, and red, respectively. These IFU's cannot be selected. To select an `Active` IFU the user should use the centre mouse button. The selected IFU is then shown with a bar beneath its numbered box.

If you drag the mouse over an IFU you will be prompted with the name of the target observed with that IFU. *Warning:* it might bee necessary to drag the mouse over the Spectrum plot (see below) to refresh the information.

At the bottom of the interactive window the extracted science frame spectrum (for the selected IFU) is displayed with wavelength units of microns. The flux is shown as a blue line, while the location of sky emission lines (OH) are marked in red. Currently, the extracted spectrum is derived in the python script of the interactive window. Each 14×14 pixel plane of the science data cube is averaged and plotted as a function of wavelength. To the right of the extracted spectrum is a median collapsed image of the reconstructed science data cube.

On the right-hand edge of the interactive window the user may modify a number of parameters to improve the science frame reduction. The parameters and their description is given in table 9.4. If parameters have been changed, then clicking on the `Re-run Recipe` will re-execute the *kmos_sci_red* pipeline routine.

Other recipe parameters, not present in Table 9.1, can be modified by opening the `Cube reconstruction` actor, double clicking on the RecipeExecuter associated to the *kmos_sci_red* recipe (the green box named *kmos_sci_red_1*), and editing the desired parameter in the apposite field.

You can re-use your preferred parameter set as initial values for subsequent recipe executions by clicking the corresponding button. The new values will be stored for this dataset, but they will be lost afterwards. To save user-customized values as default open the `Cube Reconstruction` composite actor, identify the parameter you want to change the default from the list (the `init_` parameters identified with the red bullet in the sub-workflow canvas), double-click to edit. These values will be used till the workflow is open, but they will be lost when closing it, unless you save it.

The bottom right list indicates all the files that were used as input to the interactive window, identified by their category. If the user wants to inspect a particular file or assess the recipe products with another viewer or script, (s)he can copy the full file path from the list.

9.4.1 Flux calibration

The flux calibration of the scientific data is performed by the *kmos_sci_red* recipe by applying a conversion factor using the photometric zeropoint. The zeropoint is stored in the header of the `TELLURIC`, `TELLURIC_ONLY` or `RESPONSE` files (keyword `HIEARCH ESO QC ZPOINT`). The workflow parameter **telluric and response correction** determines which file to provide to *kmos_sci_red_1*, and terefore which zeropoint to use. More information on the use of **telluric and response correction** can be found in Section 8.2.3.

The flux units of the products is $erg\,s^{-1}\,cm^{-2}\,\text{Å}^{-1}$.

Note: the flux calibration is performed only if there is a zeropoint for each detector stored in the input file (`TELLURIC`, `TELLURIC_ONLY` or `RESPONSE`). For example, when using the `TELLURIC` computed via *kmos_standard*, this condition is ensured if if the input data set contains at least three standard star observation (i.e. one per detector).

In the case the flux calibration cannot be performed, the products will be in units of ADU. In this latter case, the user can apply the flux conversion by himself. If the zeropoint of one detector is missing, the average value stored in the RESPONSE static calibration file can be used. The header keyword HIERARCH ESO QC MAD_THR contains the median absolute deviation of the zeropoints computed so far in the ESO archive, and can be used as an estimate for the zeropoint error.

The zeropoint (QC.ZPOINT) is defined so that:

$$\text{magnitude} \;=\; \texttt{QC.ZPOINT} - 2.5\, log_{10}\,(\text{counts/second})$$

To convert the SCI_RECONSTRUCTED data cubes from counts/second into physical units, the following conversion can be applied:

$$\text{Flux density} \;=\; \text{counts/second} \times F_0 \times 10^{-0.4 \times \texttt{QC.ZPOINT}}$$

where, F_0 is the zero magnitude flux density in whatever units have been used. The values of F_0 are listed in table 9.3 for a units of $W/m^2/\mu m$ and $ph/s/m^2/\mu m$. The conversion factor between $W/m^2/\mu m$ and $erg/cm^2/$Å is: [$erg/cm^2/$Å] = [$W/m^2/\mu m$]/10

If you want to derive a line flux, simply integrate the counts over the line, convert the result to a flux density, and then multiply by the spectral size of a pixel (as given by the CDELT3 keyword in the cubes, or the CDELT1 keyword in the extracted spectra).

9.4.2 Optimizing the sky removal via workflow parameters

The *kmos_sci_red* recipe triggered by the Cube Reconstruction actor automatically assigns a sky IFU to an object IFU. The adopted criteria is to use the same IFU as the object, which is closest in time. To override the automatic assignment, please refer to Section 9.4.3.

Once the Object / Sky pair is identified, the recipe proceed to remove the sky from the corresponding object IFU. To suppress sky subtraction, set -no_subtract=true. This setup will create also reconstructed sky cubes.

There are several steps involved in the sky subtraction. First, the object cube is reconstructed and aligned to a reference spectrum with OH emission lines. This includes a 2nd polynomial order fit to compensate non optimal wavelength calibration. The alignment is triggered if the dataset contains a OH_LINES reference file, which is the default case and the recommended strategy. To remove this step, deselect the OH_LINES reference file from the dataset.

Then the user has the option to re-scale the intensities of the sky emission lines measured on the sky cube to match those observed in the science cube. This option exploits the sky tweaking algorithm described in Davies et al. 2007, MNRAS, 375, 1099, that defines groups of lines with the same scaling factor. This option is triggered setting -sky_tweak=true (default). In some (rare) cases, switching off the -skytweak method gives better results than the default configuration. Note that you cannot trigger this option and set -no_subtract=true.

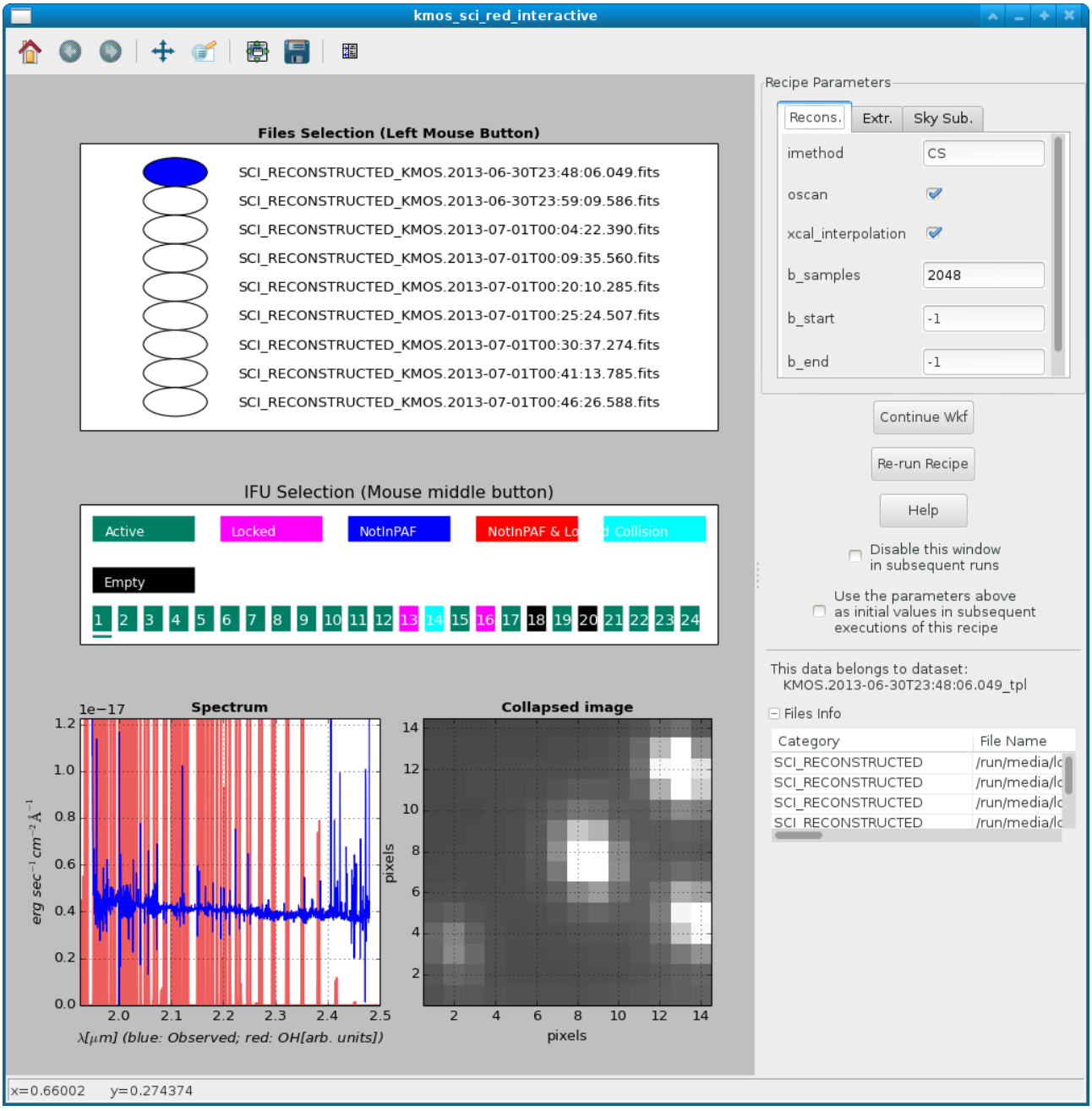


Figure 9.3: The interactive window for the `Cube Reconstruction` actor as displayed for the KMOS tutorial data set.

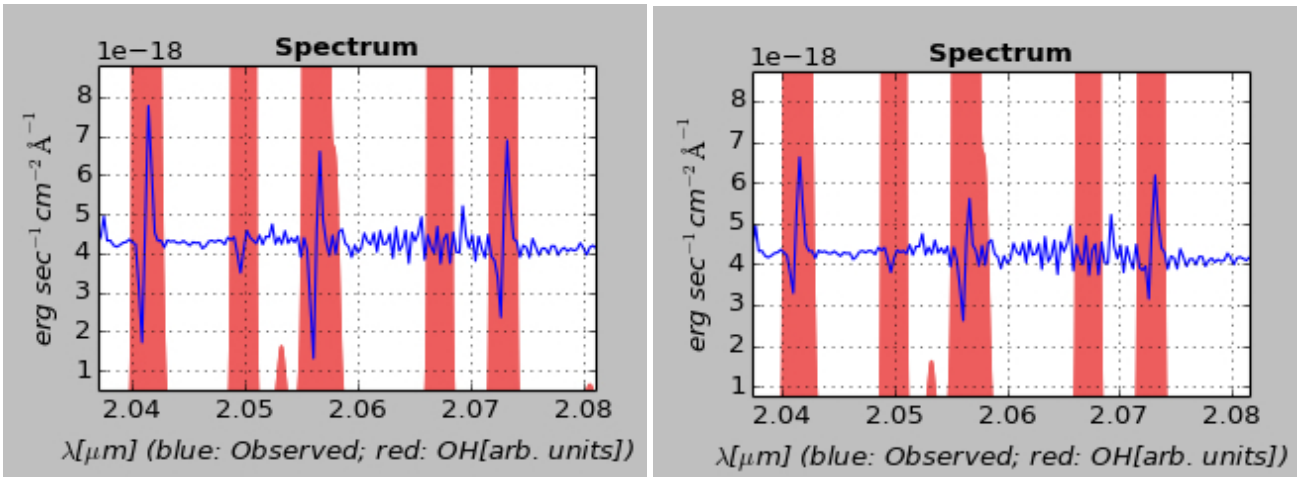


Figure 9.4: Residual sky lines with no stretching algorithm (left) and with stretching algorithm (right). The figures are zooms obtained from the interactive window of the Cube Reconstruction actor. The comparison refers to a small portion of the wavelength range, for the IFU 1 of the first exposure in the first demo dataset. The amplitude of P-Cygni profiles observed in residual skylines are diminished when using the stretching algorithm. The red stripes identify OH reference sky lines.

The sky lines scaling factors computed via the sky tweaking algorithm can be not correct if there is still a residual mismatch in wavelength between the object spectrum and the science spectrum. This scenario can be identified by the presence of P-Cygni profiles in the residual sky lines in the reconstructed datacubes. In some cases, it is possible to correct for this wavelength mismatch by applying an additional re-alignment of the skylines measured in the sky and object cubes. This can be done by setting the parameter `-stretch=true`.

The algorithm detects bright emission lines from 1D sky spectra extracted from the object and the sky datacubes, identifies the same lines in both spectra, computes the difference in position, and computes a polynomial correction to minimize these differences. This correcting polynomial is then applied to the sky cube only in order to align it better to the object cube.

Note that the alignment of the sky cube to the OH_LINES reference spectrum is still done before applying the stretching algorithm, unless `-skip_sky_ oh_align` is set to `true`. Skipping the sky alignment reduces the number of interpolations to reconstruct the sky datacube. However, in case of large offsets, it might still be useful to have this initial alignment before applying the stretching algorithm. Note that `-skip_sky_oh_align=true` has an effect only if the stretching algorithm is activated.

The outcome of the stretching algorithm needs to be evaluated with care. It is suggested to use high order polynomials (e.g. between 4 and 14) by setting the `-stretch_degree` parameter accordingly (the default degree is 8). The use of high order polynomials help in correcting for small wavelength mismatches in most of the spectral range, however it can introduce spurious effects at the edges of the wavelength range.

Figure 9.4 compares the results of the stretching algorithm on the first demo dataset (first exposure, first IFU) with default configuration. The amplitude of P-Cygni profiles observed in residual skylines are diminished when using the stretching algorithm.

9.4.3 Object / sky association for sky subtraction

The *kmos_sci_red* recipe parameter `-obj-sky-table` allows the user to specify the path to an ascii file that associates every object exposure with its corresponding sky, overriding the automatic association done by the recipe itself.

In the case the user wants to override the automatic association and feed the pipeline with his/her own, the suggested procedure is:

1. Run the workflow, selecting only one dataset and wait till the *kmos_sci_red* interactive window appears.
2. Locate the file `obj_sky_table.txt` produced by the *kmos_sci_red* recipe. It is located in the directory: `BOOK_KEEPING_DIR/kmos_sci_red/latest`.
3. Copy it into a safe place, and rename in a way that makes easy to associated it with the current dataset.
4. Change it according to the needs (an example is provided below). Each time the file is edited, it should be saved with a new name, otherwise the reflex lazy mode will ignore the changes and will re-use the previous results.
5. Enter the edited file name with its full path to the `-obj-sky-table` field in the interactive window.
6. Press `Re-run Recipe` to apply the changes. Each time a new file name is entered, Reflex will re-run the recipe. If a file has already been used (and if the other recipe parameters are unchanged), then the lazy mode will be activate and re-use the previous results without re-running the recipe. It is recommended not to save this parameters as new default, otherwise the other datasets will also use this object-sky association file.

9.4.4 Example of object/sky association table.

The ascii file that associates every object exposure with its corresponding sky looks like the following (the caption lines are not shown here, for sake of clarity).

```
# [caption lines skipped]
Object/sky associations of frames tagged as: SCIENCE

index: filename:
#  0:  reflex_input/kmos-demo-reflex-1.1/raw/KMOS.2013-06-30T23:48:06.049.fits
#  1:  reflex_input/kmos-demo-reflex-1.1/raw/KMOS.2013-06-30T23:53:23.571.fits
#  2:  reflex_input/kmos-demo-reflex-1.1/raw/KMOS.2013-06-30T23:59:09.586.fits
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
IFU          1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18   19   20 21 22 23 24
            ---------------------------------------------------------------------------
frame #  0:  reflex_input/kmos-demo-reflex-1.1/raw/KMOS.2013-06-30T23:48:06.049.fits
      type:  O  O  O  O  O  O  O  O  O  O  O  O  .  .  O  .  O  O    O    O  O  O  O  O
  sky in #:  1  1  1  1  1  1  1  1  1  1  1  1  .  .  1  .  1  1    1    1  1  1  1  1
frame #  2:  reflex_input/kmos-demo-reflex-1.1/raw/KMOS.2013-06-30T23:59:09.586.fits
      type:  O  O  O  O  O  O  O  O  O  O  O  O  .  .  O  .  O  O    O    O  O  O  O  O
  sky in #:  1  1  1  1  1  1  1  1  1  1  1  1  .  .  1  .  1  1   1/20  1  1  1  1  1
-------------------------------------------------------------------------------------------
```

The IFUs pointing to Object and Sky are identified with the letters "O" and "S", respectively. Inactive arms are identified with a dot ".". For each frame and IFU containing an object, the corresponding frame and IFU containing the sky is displayed in the line below. If the sky location is described with a single number, e.g. N (N is an integer), then the sky to be used is on the exposure with index N, on the same IFU as the object.

If the sky location is described in the format N/M (N and M integers, with $1 < M < 24$), then the sky to be used is located in the exposure with index N and IFU M.

In the example shown above, in the object frame KMOS.2013-06-30T23:48:06.049.fits (that has index 0) each n-th IFU has the corresponding sky cube in the same n-th IFU of the frame with index 1 (KMOS.2013-06-30T23:53:23.571.fits)

The same is true for the object frame KMOS.2013-06-30T23:59:09.586.fits (that has index 2), except for IFU 19. The corresponding sky has to be found in the frame with index 1, at ifu 20.

9.5 Science reduction: datacube combination

The next step in the data reduction is the combination of the reconstructed cubes that belong to the same target, or to reconstruct the full mosaic in the case of mosaic observations. This is automatically done in the `Cubes Combination` actor.

This actor is identifiable by an orange rectangle encompassing the actor name. The interactive mode is enabled by default. Should you wish to change that use `Open Actor` to get access to the the components of the interactive actors, then double-click on the composite actors, setting the "EnableInteractivity" parameter to `false`, and clicking `Commit` to save the changes to the workflow.

Figure 9.5 shows the interactive window that will pop-up at the end of the execution of the *kmos_combine* pipeline recipe (for the first demo-dataset).

The file list on the top-left panel ("Files selection") shows the names of the combined datacubes produced by the recipe. The user can select the desired output file name to inspect by double-clicking wiht the left mouse button (it might require few seconds for the other windows to refresh).

The right window ("File Spectra") displays the spectra of the reconstructed cubes that contributed to the creation of the selected combined datacube. This window might not be available for observations taken in mosaic mode. These spectra are obtained by integrating the datacube along the 2 spatial direction. These are the spectra of the various extensions of the `SCI_RECONSTRUCTED` datacubes, which are combined into the selected datacube. On the right part of the plot, a code n:m for each spectrum is given, specifying the raw file number (n) and the IFU number (m). If one of the input spectra to be combined displays significant deviations from the others, it might be worth considering to remove if from the combination. To remove some datacubes from the combination, insert the corresponding n:m codes in the `skipped_frames` recipe parameter. Spectra of excluded frames will be grayed out in the File Spectra window.

The bottom left plot ("Spectrum") shows in blue the spectrum of the selected combined datacube (obtained by integrating the datacube along the 2 spatial direction). It also shows in red the location of the sky emission lines (OH). The bottom right plot ("Collapsed image") shows the reconstructed field of view of the selected datacube, obtained by integrating the datacube along the wavelength direction. For mosaic observations, this will show the full mosaic field of view.

On the right-hand edge of the interactive window the user may modify a number of parameters to improve the science frame reduction. The parameters and their description are given in table 9.5. If parameters have been changed, then clicking on the Re-run Recipe will re-execute the *kmos_combine* pipeline routine.

You can re-use your preferred parameter set as initial values for subsequent recipe executions by clicking the corresponding button. The new values will be stored for this dataset, but they will be lost afterwards. To save user-customized values as default open the `Cube Combination` composite actor, identify the parameter you want to change the default from the list (the `init_` parameters identified with the red bullet in the sub-workflow canvas), double-click to edit. These values will be used till the workflow is open, but they will be lost when closing it, unless you save it.

The bottom right list indicates all the files that were used as input to the interactive window, identified by their category. If the user wants to inspect a particular file or assess its quality with another viewer or script, (s)he can copy the full file path from the list.

At the bottom of the interactive window, the user can select 3 visualization modes. The first (default) shows the window as in Figure 9.5. The second shows the list of output spectra (the "Files Selection" plot), the spectrum (the "Spectrum" plot) and the reconstructed field of view (the "Collapsed Image" plot), of the selected output datacube. The last option shows only the input spectra (the "File Spectra" plot) that contributed to the creation of the selected datacube.

9.5.1 Combining exposures from different Observing Blocks

By design, the KMOS workflow combines the exposures associated to a single observing block (OB). If the user wants to combine the exposures of multiple OBs (s)he has to run the *kmos_combine* pipeline recipe on the workflows products manually. The suggested procedure is the following:

- Reduce all the OBs to be combined with the reflex workflow. The products to be combined (i.e., the `SCI_RECONSTRUCTED` frames) will be saved into the directory specified in `END_PRODUCTS_DIR` (see Section 8).
- Create the set of frames file (SOF). It is an ascii file that specifies all the inputs of the pipeline recipe to be run and their categories. In the following example, the data.sof file contains the list of all the observations taken in 3 different OBs:

```
reflex_end_products\dataset1\SCI_RECONSTRUCTED_1.fits SCI_RECONSTRUCTED
reflex_end_products\dataset1\SCI_RECONSTRUCTED_2.fits SCI_RECONSTRUCTED
reflex_end_products\dataset1\SCI_RECONSTRUCTED_3.fits SCI_RECONSTRUCTED
reflex_end_products\dataset2\SCI_RECONSTRUCTED_1.fits SCI_RECONSTRUCTED
reflex_end_products\dataset2\SCI_RECONSTRUCTED_2.fits SCI_RECONSTRUCTED
reflex_end_products\dataset3\SCI_RECONSTRUCTED_1.fits SCI_RECONSTRUCTED
reflex_end_products\dataset3\SCI_RECONSTRUCTED_2.fits SCI_RECONSTRUCTED
reflex_end_products\dataset3\SCI_RECONSTRUCTED_3.fits SCI_RECONSTRUCTED
reflex_end_products\dataset3\SCI_RECONSTRUCTED_4.fits SCI_RECONSTRUCTED
```

  Note that the data.sof must contain the full path to the file, or the relative path from the command line execution.

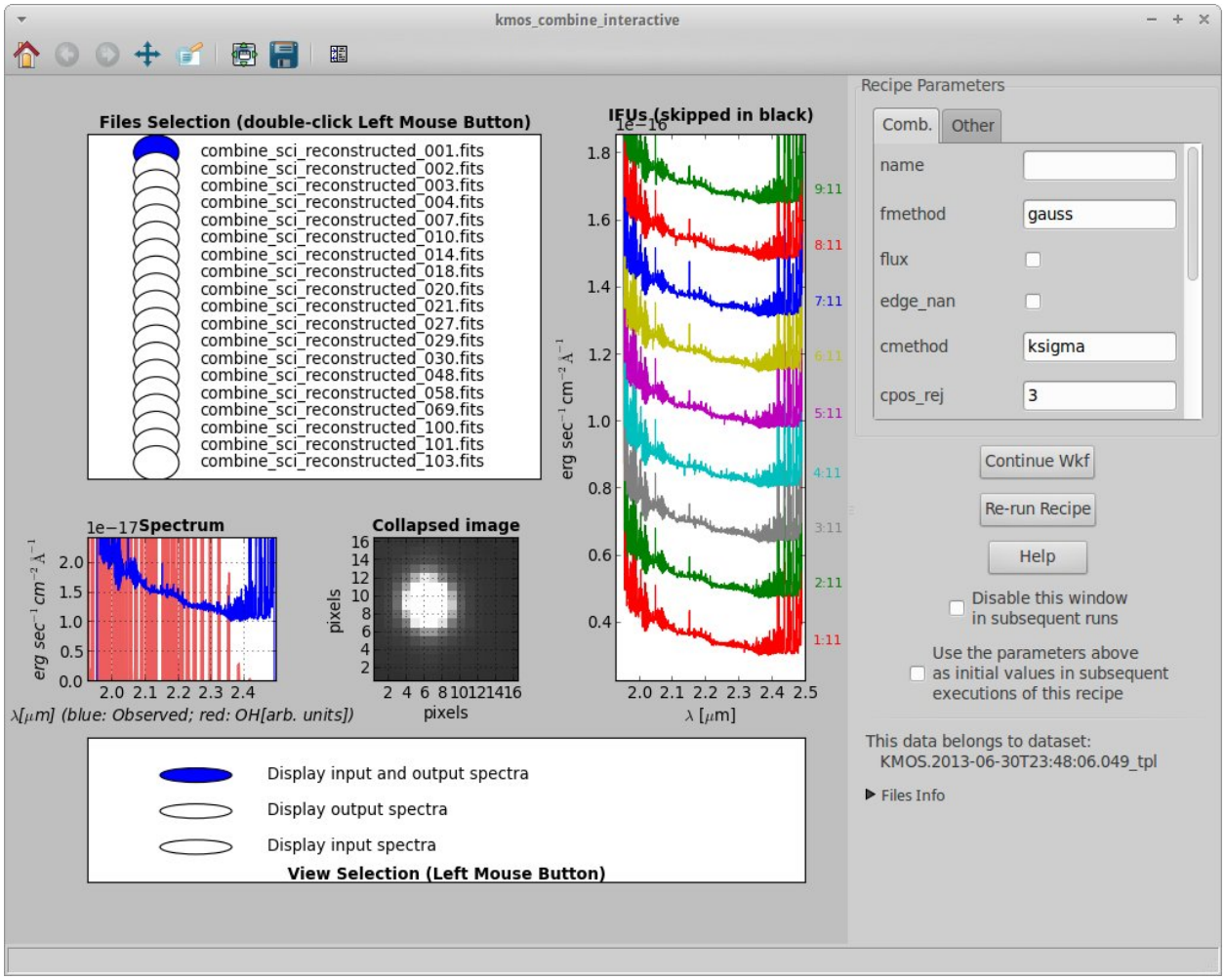


Figure 9.5: *The interactive window for the* `Cube Combination` *actor as displayed for the KMOS tutorial data set.*

- Execute the pipeline recipe *kmos_combine* with the following command line.

```
esorex kmos_combine data.sof
```

For a full list of the recipe parameters and inputs/outputs, please consult the KMOS pipeline Manual. It is recommended to combine data of the same exposure time. If one

Table 9.2: Parameters that a user can manipulate within the `Standard_Star` interactive window

Reconstruction		
Parameter[1]	Value (default)	Explanation
–imethod	`CS`	The interpolation method used for reconstructing the standard star data cube. Note that no error spectra will be generated for this interpolation method. The full list of interpolation methods include: **CS**: Cubic-spline (default). Warning: this method does not compute the errors on the telluric correction. **NN**: Nearest Neighbour. **lwNN**: linear-weighted Nearest Neighbour. **swNN**: square-weighted Nearest Neighbour. **MS**: modified Shepard's method.
–xcal_interpolation	`TRUE`	If **TRUE** interpolate the pixel position in the slitlet (xcal) using the two closest rotator angles in the calibration file. Otherwise take the values of the closest rotator angle.
–b_samples	`2048`	The number of samples in wavelength for the reconstructed cube
–b_start	`-1`	The lowest wavelength [μm] to take into account when reconstructing (default of `-1` sets the proper value for the actual band automatically)
–b_end	`-1`	The highest wavelength [μm] to take into account when reconstructing (Default of `-1` sets the proper value for the actual band automatically)
Extraction		
–fmethod	`gauss`	The type of function that should be fitted spatially to the collapsed image. This fit is used to create a mask to extract the spectrum of the object. Valid optios are **gauss** and **moffat**
–mask_method	`optimal`	Method used to extract stellar spectrum. Valid entries are: **mask**. Pixels to extract the stellar spectrum are identified via user-imput mask. **optimal** (default). Center and aperture are automatically defined. **integrated**. Center and aperture are set by recipe paramters.
–centre	7.5.,7.5	The center of the circular mask (in pixels). If center=-1, then the center is automatically searched for each IFU.
–radius	3.0	The radius (in pixels) for the integrated method.
–neighborhoodRange	`1.001`	Defines the range to search for neighbors during reconstruction.
–flux	`FALSE`	Apply conservation of flux (**TRUE/FALSE**).
Combination		
–cmethod	`ksigma`	The method of frame combination to be used. **ksigma** is an iterative sigma clipping. For each position all pixels in the spectrum are examined. If they deviate significantly, they will be rejected according to the conditions: val $>$ mean + stdev * cpos_rej and val $<$ mean - stdev * cneg_rej The full list of combination methods include: **ksimga**: kappa-sigma clipping **median**: at each pixel position the median is calculated **average**: at each pixel position the average is calculated **sum**: at each pixel position the sum is calculated **min-max**: The specified number of minimum and maximum pixel values will be rejected. –cmax and –cmin apply to this method
–cpos_rej & –cneg_rej	`3.0 & 3.0`	The positive and negative rejection thresholds for bad pixels.
–citer	`3`	The number of iterations for kappa-sigma-clipping (applies only when –cmethod = `ksigma`).
–cmax & –cmin	`1 & 1`	The number of maximum and minimum pixel values to clip with min/max-clipping (applies only when –cmethod = `min_max`).
Star		
–startype		If empty, the information is read from the file header. Otherwise, header information is overridden. This value applies to all objects examined in the input frames. Examples could be **A3I**, **G3IV** or **K0I**. The first letter defines the star type, the second letter the spectral class and the last letters the luminosity class.
–magnitude		If empty, the information is read from the file header. If this parameter is specified, the stored magnitudes in the FITS headers are overridden. For HK two magnitudes for each H and K have to be specified (separated by a comma). All other gratings use a single magnitude.

Table 9.3: Zero Magnitude Flux Densities (from [3]).

KMOS band	2MASS band	Band pass for calibration (μm)	Zero magnitude flux density (F_0)	
			$W/m^2/\mu m$	$ph/s/m^2/\mu m$
K	K	2.028 – 2.290	4.283×10^{-10}	4.65×10^{9}
HK	H & K	1.5365 – 1.7875 & 2.028 – 2.290	1.133×10^{-9} & 4.283×10^{-10}	9.47×10^{9} 4.65×10^{9}
H	H	1.5365 – 1.7875	1.133×10^{-9}	9.47×10^{9}
YJ	J	1.154 – 1.316	3.129×10^{-9}	1.944×10^{9}
IZ	—	0.985 – 1.000	7.63×10^{-9}	3.81×10^{10}

Table 9.4: Parameters that a user can manipulate within the `Cube Reconstruction` interactive window

Reconstruction		
Parameter[1]	Value (default)	Explanation
–imethod	`CS`	The interpolation method used for reconstructing the science data cube. A Cubic-spline (`CS`) interpolation is used as the default. Note that no error spectra will be generated for this interpolation method. The full list of interpolation methods include: **CS**: Cubic-spline **NN**: Nearest Neighbour **lwNN**: linear-weighted Nearest Neighbour **swNN**: square-weighted Nearest Neighbour **MS**: modified Shepard's method
–oscan	`TRUE`	If **TRUE**, it applies a level correction of the detector channels.
–xcal_interpolation	`TRUE`	If **TRUE** interpolate the pixel position in the slitlet (xcal) using the two closest rotator angles in the calibration file. Otherwise take the values of the closest rotator angle.
–b_samples	`2048`	The number of samples in wavelength for the reconstructed cube
–b_start	`-1`	The lowest wavelength [μm] to take into account when reconstructing (the default of `-1` sets the proper value for the actual band automatically).
–b_end	`-1`	The highest wavelength [μm] to take into account when reconstructing (the default of `-1` sets the proper value for the actual band automatically).
–obj_sky_table		Specify the path to an ascii file with the object-sky association (see Section 9.4.3).
Extraction		
–fmethod	`gauss`	The type of function that should be fitted spatially to the collapsed image. This fit is used to create a mask to extract the spectrum of the object. The full list of fitting methods include: **gauss** and **moffat**
–neighborhoodRange	`1.001`	Defines the range to search for neighbors during reconstruction.
–flux	`FALSE`	Apply conservation of flux (**TRUE/FALSE**).
Sky Sub.		
–smethod	`FALSE`	The interpolation method used for shifting. The full list of shifting methods include: **CS**: cubic spline **NN**: nearest neighbour
–background	`FALSE`	Specify if a background subtraction should be applied. The backround removed by this option includes also object continuum. The background is evaluated in the spectral regions free of intense skylines. This option can be used, for example to better isolate emission features in the science target from its continuum.
–no_subtract	`FALSE`	If set to `TRUE`, the found objects and references will **not** be sky subtracted. Additionally all IFUs will be reconstructed, even the ones containing skies. This option sets the parameter no_combine to TRUE automatically.
–sky_tweak	`TRUE`	If set to `TRUE`, it activates the sky tweaking algorithm for sky subtraction.
–velocity_offset	`0.0`	It specifies velocity offset correction in km/s for wavelenth scale.
–stretch	`FALSE`	Stretch sky before sky tweaking.
–stretch_degree	`8`	Stretch polynomial degree.
–stretch_resampling	`spline`	Stretch resampling method (linear/spline).
–skip_sky_oh_align	`FALSE`	If `-stretch=true`, it skip the alignment of the skycube to the reference `OH_LINES` spectrum.

[1] Only the most important parameters available to the *kmos_sci_red* pipeline routine, have been included for use in the `Cube Reconstruction` interactive window.

Table 9.5: Description of the `kmos_combine` recipe parameters.

Parameter[1]	Value (default)	Explanation
–name		: Name of the object to combine. Empty string means to combine all.
–fmethod	gauss	The fitting method for image centroid computation (applies only when method='center'): "gauss": fit a gauss function to collapsed image (default), "moffat": fit a moffat function to collapsed image.
–ifus		: The indices of the IFUs to combine. "ifu1;ifu2;...".
–method	header	: The shifting method: 'none': no shifting, combined directly, 'header': shift according to WCS (default) 'center': centering algorithm, 'user': read shifts from file.
–filename		: The path to the file with the shift vectors (method='user').
–flux	false	: Apply flux conservation: (TRUE (apply) or FALSE (don't apply).
–edge_nan	false	: Set borders of cubes to NaN before combining them.(TRUE (apply) or FALSE (don't apply).
–skipped_frames		: List of indexes of the frames to skip. Format: n1:m1, n2:m2, ...; n is the n-th RAW input frame; m is the m-th IFU. The syntax n:m means that the m-th IFU of the n-th input frame will be ignored. Empty string (default) means do not exclude anything.
–suppress_extension	false	: Suppress arbitrary filename extension.
–cmethod	ksigma	: Apply "average", "median", "sum", "min_max." or "ksigma".
–cpos_rej	3	: The positive rejection threshold for kappa-sigma-clipping (sigma).
–cneg_rej	3	: The negative rejection threshold for kappa-sigma-clipping (sigma).
–citer	3	: The number of iterations for kappa-sigma-clipping.
–cmax	1	: The number of maximum pixel values to clip with min/max-clipping.
–cmin	1	: The number of minimum pixel values to clip with min/max-clipping.

10 Tips

In this Section we present some suggestions that can help the user to improve some aspects of the data reduction. This section is in constant change, and it will be updated while we gain experience in the data reduction and add functionality to the pipeline.

The following suggestions work in some cases, but there is no guarantee that they are valid for all the datasets.

10.1 Dealing with residual sky emission lines.

One of the major challenges with the KMOS data reduction is sky subtraction. In particular, residual of sky emission lines can be difficult to remove. Section 9.4.2 describes how to optimize the sky subtraction using the workflow interactive window. Here we list some strategies that can be applied to the reduced cubes produced by the reflex workflow.

- In Mosaic mode, if there are IFUs that contain negligible contamination form science sources, the following post-processing strategy can reduce the amplitude of residual sky lines.

  1. Run the workflow with your favourite recipe parameters, and set `-background` to "true" in the Cube Reconstruction actor. In this way, we remove at best the underlying continuum. Let's refer to the combined final datacube (PRO.CATG = `COMBINE_SCI_RECONSTRUCTED`) obtained with this technique as "optimized residual".
  2. Run the workflow with your desired parameters, and set `-background` set to false in the Cube Reconstruction actor. Let's refer combined final datacube obtained with this technique as "sky removed".
  3. In the "optimized residual" datacube, locate a portion in the field of fiew corresponding to one IFU where the science target contamination is minimal. Construct a single spectrum from these regions by averaging the spectra in the selected region. This represents the "spectrum of residuals".
  4. Subtract the "spectrum of residuals" from the "sky removed" datacube.

  It is worth noting that with this technique, we do not subtract directly the sky lines evaluated on a single IFU. Indeed, sky lines can vary in different IFUs, and they are subtracted in step 2. We subtract "the residuals", and the results are as good as long as the residual have a common structure among the IFUs (which is sometimes the case).

- For isolated and small sources (much smaller than the IFU size) it might be worth trying to evaluate the sky (or the sky residuals) to subtract on the spaxels around the scientific source. To skip the sky subtraction in the KMOS workflow, set `no_subtract = true` in the Cube Reconstruction actor

- For science sources that have spectral features more broad than than the typical size of sky residuals (which are few pixels), one can try a spectral median filtering for each spectrum in the reconstructed datacube. The value at wavelength of pixel coordinate z is compared to the background level b estimated as the median of the spectrum in the wavelength regions $z-2\cdot dz < z < z-dz$ and $z+dz < z < z+2\cdot dz$. If the value at z differs from the median more than $k \cdot \sigma$, where σ is the standard deviation evaluated in wavelength region specified above, then it is replaced by b. The following values are good for targets that

have spectral features with FWHM larger than ~ 15 pixels: $dz = 3$, $k = 3$. It is advisable to check the object spectral features before and after the filtering to evaluate whether they have been deteriorated by the filtering procedure.

10.2 Process data when OBJECT and SKY are taken from two separate Observing Blocks

It is always recommended to observe object and sky within the same observing block. If this is done, the kmos workflow will include object and sky within the same dataset and process them together. If sky and object are from different OBs, they will be listed into two different datasets. In order to process your data you have to process these two datasets separately, without applying the sky subtraction. You can subtract the sky later, with the kmo_arithmetic or kmos_sky_tweak recipes via the `esorex` command line interface. To do so, execute the following steps:

1. Double-click on the Cube Reconstruction actor (`kmos_sci_red` recipe) and specify the following parameters:

   ```
   init_no_subtract = TRUE
   init_sky_tweak = FALSE
   ```

2. Run the workflow on the OBJECT dataset. Save the resulting `SCI_RECONSTRUCTED` files in a separated directory (e.g., object_cubes)

3. Run the workflow on the SKY dataset. Save the resulting `SCI_RECONSTRUCTED` files in a separated directory (e.g., sky_cubes).

4. If you want to remove the sky without applying the sky tweaking algorithm, execute the `kmos_arithmetic` recipe:

   ```
   esorex kmo_arithmetic --operator='-' SCI_RECONSTRUCTED_object.fits
               SCI_RECONSTRUCTED_sky.fits
   ```

   where `SCI_RECONSTRUCTED_object.fits` `SCI_RECONSTRUCTED_sky.fits` are the reduced and reconstructed datacubes of the OBJECT and SKY observation, respectively.

5. If you want to remove the sky and applying the sky tweaking algorithm (suggested) do as follows. Create a SetOfFrames file (e.g., data.sof) with an editor; the sof file must contain 1 object cube (not sky subtracted) and 1 sky cube (not sky subtracted). An example of the content of such data.sof file is:

   ```
   object_cubes/SCI_RECONSTRUCTED_object.fits OBJECT_CUBE
   sky_cubes/SCI_RECONSTRUCTED_sky.fits SKY_CUBE
   ```

   where `SCI_RECONSTRUCTED_object.fits` `SCI_RECONSTRUCTED_sky.fits` are the reduced and reconstructed datacubes of the OBJECT and SKY observation, respectively.

   Now, remove the sky from the object cube by executing:

```
esorex kmos_sky_tweak data.sof
```

The above instructions work if the object is at the same IFU number as the associated sky. If not, one has to extract the $n-$th IFU from the object and the corresponding $m-$th IFU from the sky and process them individually. The pipeline recipe *kmo_fits_strip* can be used to extract a given IFU from the `SCI_RECONSTRUCTED` cubes.

10.3 Improving the resolution of spectral line

Sometimes it is possible to improve the spectral resolution on the final combined datacube, and therefore to better sample the profile of the lines of the science object, by increasing the sampling in wavelength.

One can try to increase the `-b_samples` parameter from 2048 (default) to 3075 (i.e., increase it by a factor 1.5). This will improve the sampling in wavelength and can result in a better spectral resolution in the final combined datacube (`COMBINE_SCI_RECONSTRUCTED`).

The `-b_samples` parameter must be the same for the Standard Star (recipe *kmos_std_star*), and Cube Reconstruction (recipe *kmos_sci_red*) actors. For other actors it can be different, although we recommend to keep it the same for consistency.

10.4 Using telluric corrections obtained with MOLECFIT

If telluric standard stars are present in the dataset, the workflow triggers the pipeline recipe *kmos_standard* and generates a file with category `TELLURIC` that include the response and telluric corrections merged together. By default, this file is passed to the science recipe for further processing.

However, the user might want to explore alternative methods to correct for telluric feature. ESO provides a tool for the removal of telluric absorption named `molecfit`, which can be run independently of any other ESO pipeline. Further information on `molecfit`, manuals, its installation and use can be found at the following link:

http://www.eso.org/sci/software/pipelines/skytools/molecfit.

Please cite the following paper in case you use `molecfit` in your data reduction process: Smette et al., 2015 A&A, 576, 77.

The advantages of using this tool are to efficiently remove telluric sky lines from scientific data with: no necessity to take into account stellar features present in B5 and colder stars, bad continuum reconstruction, and it is not prone to noise features. Also, it is useful to avoid time-consuming observations of with 24 arms sciencepattern standard star template, indeed `molecfit` can be applied idependently to all the KMOS arms as soon as the spectrum has a well detected continuum. The tool can be used with both the telluric standard observations or directly with scientific targets.

Currently `molecfit` can process 1D input spectra and is not directly integrated in the `Reflex` workflow. Aim of the following sections is to provide the user with step-by-step instruction to use the telluric correction provided by `molecfit` when processing KMOS data with the `Reflex` KMOS workflow.

10.4.1 Correction for response curve and telluric features

The TELLURIC file produced by the KMOS pipeline contains a correction spectrum that combines the instrument response the telluric absorptions. In the case the user wants to correct for telluric absorptions a posteriori with molecfit, it is possible to set the workflow to skip the telluric correction, and leave just the response correction. This can be easily done by setting the workflow parameter **telluric and response correction=2**.

In this case the correction for instrument response will be performed using a static calibration file (category: RESPONSE), and the flux calibration will be performed by using the header information of the TELLURIC file produced by the workflow when processing the standard stars.

The response correction in the static calibration file has an error of $\sim 5\%$ (in the regions not affected by telluric absorptions) up to $\sim 15\%$ (in the regions most affected by telluric absorptions).

10.4.2 Using molecfit

As described above, the best way to use molecfit to remove telluric absorption on KMOS spectra is first to run the workflow skipping the the telluric correction, and leave only the response correction and flux calibration. This can be easily done by setting the workflow parameter **telluric and response correction=2**.

Then, the telluric correction can be computed in two ways.

1. On science data directly. This is recommended in the case of high S/N. The workflow datacubes products SCI_RECONSTRUCTED are flux calibrated and corrected for instrument response. The user can then extract 1D spectra from the science datacubes (e.g., using the pipeline recipe *kmos_extract_spec*) and use molecfit to compute the telluric correction.

2. On the observed stars. The 1D extracted profiles corrected for instrument response (category STD_STAR) are stored in the reflex temporary products (as specified by the parameter TMP_PRODUCTS_DIR in the main reflex canvas), inside the PythonActorResponseCorrection/<time-stamp>/ sub-directory. The link to products of the latest execution can be found in the book keeping directory (as specified by the parameter BOOKKEEPING_DIR in the main reflex canvas), under the sub-directory PythonActorResponseCorrection/latest/products_dir. Be aware that a file of the same category is presented also among the products of *kmos_standard*, but this latter is not corrected for response.

In both cases, the files needs to be converted in a valid format for molecfit. This will be explained in the next Section.

10.4.3 Create input fits files for `molecfit`

Molecfit can produce a sky absortion spectrum once it is fed with a one-dimensional spectrum. It is therefore necessary to extract monodimensional spectra from the KMOS datacubes and adapt the header. The following procedure can be run using spectra extrated both from telluric standards or scientific exposures.

- Locate the STD_STAR produced by the the workflow. Alternatively, extract 1D spectrum from a reconstructed datacube (not telluric corrected), if the telluric correction has to be evaluated directly on the scientific data.
- Create a single fits file to feed molecfit for any useful extension of STD_STAR file that contains data, or for any of the 1D spectra extracted from the science datacube. The new fits files have single extension. The header has to be a copy of the primary header of STD_STAR or SCI_RECONSTRUCTED and must contain also the WCS information (such as starting wavelength) stored in the header extesion of the original STD_STAR or SCI_RECONSTRUCTED file.

10.4.4 Create a new TELLURIC_ONLY product and integrate it in the REFLEX workflow

For a detailed description on how to use the molecfit gui and to run a fit, we refer to the relative manuals. Were we describe the process of how to use the molecfit output to correct KMOS data.

Molecfit produces a fits file that contains the atmospheric transmission, that ends in _TRA.fits. The exact naming and location are specified in the molecfit configuration input file.

The _TRA.fits can be used into two ways.

- Direct correction (recommended). One can simply divide the datacube of a specific IFU by the molecfit output. The user has to pay attention to associate the molecfit output to the proper file and extension. It is the fastest method in the case of low number of exposures to correct.
- Correction via the KMOS workflow (for expert users or correction of multiple datasets). One can construct a file that contains the molecfit outputs in the appropriate extensions, with category TELLURIC_ONLY. This file can be fed back to the KMOS workflow in the input data directory. To use this file, re-run the workflow with **telluric and response correction = 3**. In this way, the workflow will combine the information in TELLURIC_ONLY and RESPONSE when processing the scientific data. This fits file needs to have the same structure of the static calibration file RESPONSE. Be sure that the primary header contains the exact category (TELLURIC_ONLY), each (useful) extension contains the molecfit transmission (the molecfit fits product that ends with TRA.fits), and that each (useful) header extension contains the correct information of the zeropoint (stored in the STD_STAR files). Another important aspect of the primary header of TELLURIC_ONLY is that it needs to contain the keyword of the observing template ID (HIERARCH ESO TPL ID) , and the keywords HIERARCH ESO PRO STDSTARn = 1 for all and only the n useful IFUs. For example, in the case the telluric correction is available for IFUs 3, 12, and 19 (detectors, 1, 2, and 3) the primary header shoud contain:

  ```
  HIERARCH ESO TPL ID = 'KMOS_spec_cal_stdstar' / Template signature ID
  HIERARCH ESO PRO STDSTAR3 = 1
  HIERARCH ESO PRO STDSTAR12 = 1
  HIERARCH ESO PRO STDSTAR19 = 1
  ```

  This method is suggested if a given correction has to be applied to a large number of datasets.

In the next sessions we provide some insight on how to use molecfit on KMOS data.

10.4.5 Run `molecfit`

For a detailed description on how to use the `molecfit` gui and to run a fit, we refer to the relative manuals. Here we lists a series for general suggestion to perform a good fit of the KMOS data:

- Inclusion region: it is recommended to restrict the fits to those regions where major contributions are expected, but anyhow to include in the fit region with continuum signal. This increase the computation velocity and allow to apply low order polynomials to fit the continuum. Note that the continuum of each inclusion region is fitted independently.
- Exclusion regions: it is important to exclude from the fit spectral features belonging to the scientific source.
- Molecule selection: it is suggested to perform a step by step fitting whenever the waveband of interest allows it. Start to fit spectral region where only a single molecule is dominating the atmospheric transparency. If a weaker features should be fitted, use the derived value to determine the trace gas contributions in a second run, incorporating the results of the first run without fit option to the strong absorber. Extend the procedure the entire spectral range. It is not mandatory to fit all the possible molecules. The user can choose to include only the main contributors to the fit, and then include other molecules in the model. For the lasts, the relative abundance is computed on the bases of the standard atmosphere model.
- Atmospheric profile: automatic retrieval of two atmospheric profiles from the Global Data Assimilation System (GDAS) web site is currently not working. Anyhow for Parnal Observatory, the tool has an internal library of atmospheric profiles and related parameters, such as the precipitate water vapor, can be left free to vary.
- Wavelength correction: `molecfit` has robust and sophisticated algorithm to match the observed spectrum with the known position of the atmospheric features. For the tests conducted till now, such task is not required for KMOS as the wavelength calibration provided by the pipeline is accurate enough.
- Resolution and fitting kernel: the line spread function (LSF) of the Sectrograph strongly influences the fit quality (see the `molecfit` user manual for details). In `molecfit` it is possible to convolve the model with three different profiles to determine the LSF: a boxcar, a Gaussian and a Lorenzian profile. To avoid complicated kernel and unrealistic fit results the user should fix the individual fit component to reduce the number of degrees of freedom. Alternatively the user can select directly a single Voigt profile kernel or a kernel whose size in pixels linearly increases with wavelength. Finally, the user can provide own LSF in the form of an ASCII table of fixed kernel elements. In a multi-IFU instrument like KMOS the characterization of the LSF can be complicated (it varies from IFU to IFU and depends e.g. on the rotator position) and the way to feed it to `molecfit` can change depending on the scientific goal. In Section 3.1 the KMOS LSF is discussed.
- It is needed to have the products of `molecfit` defined on the same wavelenght vector as the products of the kmos pipeline.

10.4.6 Characterization of the Line Spread Function

KMOS LSF is both a function of the spatial and spectral position. Such trends for the H-band gratings are shown in Fig.10.1. Moreover since the instrument is mounted at a Nasmyth focus of the VLT it is equiped with

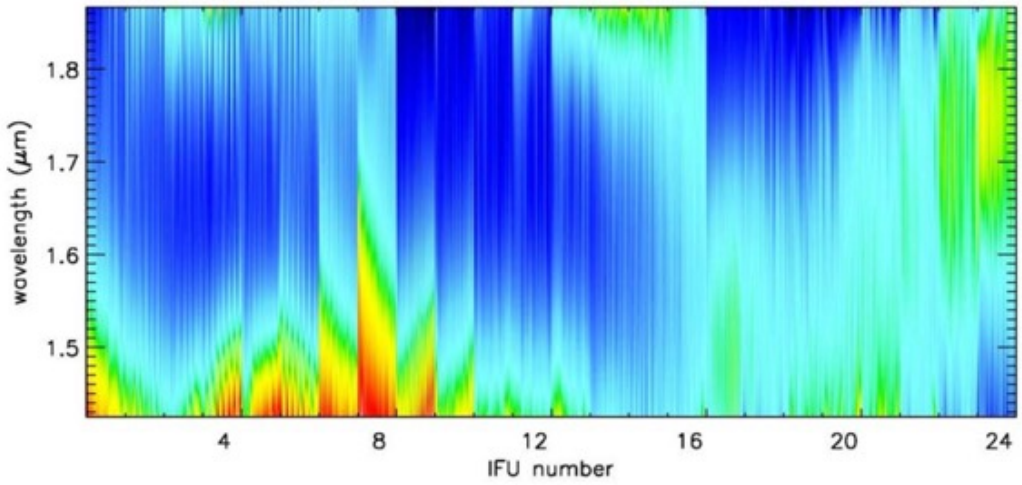


Figure 10.1: LSF map of KMOS in H-band as a function of spatial and spectral position. Slitlets from the IFUs (labelled 1-24) have been drawn side-by-side. Resolution is indicated by colour from 3Å(dark blue) to 6.7Å(red).

a rotator and flxures are therefore inevitable.

10.4.7 How to estimate the KMOS response function

The current KMOS pipeline distribution has a static calibration file that contains the average instrument responses and zeropoint for all the IFUs (category: `RESPONSE`). The calibration is accurate from 5% (in the regions not affected by telluric absorptions) up to 15% (in the regions most affected by telluric absorptions).

In the following, we describe how to compute a custom response curve, specific for a given set of observations. Because it is very difficult to disentangle the contribution of instrument response and telluric atmospheric absorption, the procedure will not return a much better precision that what provided by the calibration files. However, in rare cases, it might be needed to create a custom response if the continuum shape of the target suggest so.

The product of the recipe *kmos_std_star* telluric.fits contains both the information on the telluric absorption and the wavelength dependence of the response curve. On the other hand, the telluric correction produced by molecfit contains only the information on the telluric absorption.

It is therefore possible to extract the instrument response contribution by dividing the telluric.fits file by the molecfit transmission. The following example refers to the case of one standard star spectrum. It has to be repeated for all the spectra included in telluric.fits file.

- Get the data in one valid extension of the telluric.fits produced by `kmos_std_star` (e.g., extension number 5, that refers to detector number 1), and the corresponding telluric correction obtained by `molecfit` for the same star and extension. Let's refer to these quantities by t_kmos and t_molecfit
- If needed, resample t_molecfit to the same wavelength vector as t_kmos.

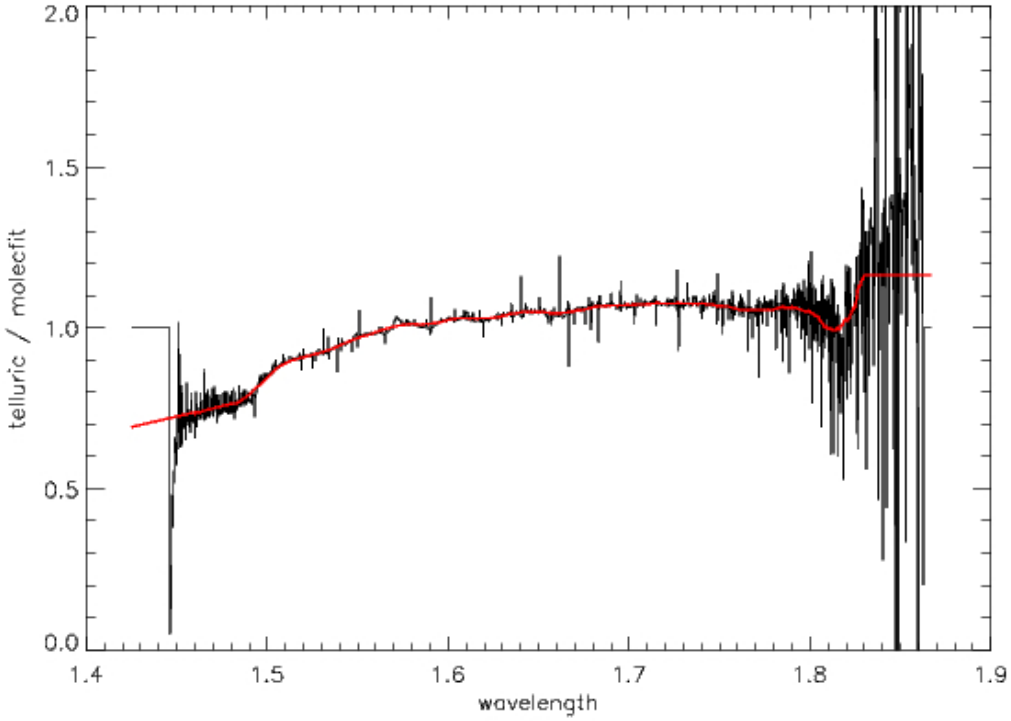


Figure 10.2: Example of the ratio between an extension of the telluric.fits file produced by the KMOS pipeline and the telluric correction produced by molecfit. The red curve, obtained by smoothing this ratio with a 200 pixels wide box car average (smooth function in IDL), represent the instrument response for that IFU.

- Compute the ratio t_kmos / t_molecfit. An example of such a ratio is shown in Figure 10.2 . The large-scale trend of this curve represents the instrument response curve, whereas the high frequency structures, i.e. the spikes in the curve, represent the ratio between the telluric corrections by the kmos pipeline and molecfit.
- Isolate the general trend of the ratio computed above, i.e. by applying a median filter or kernel smoothing. It might be necessary to extrapolate the results for the edges, where the noise is higher. The general trend represent the instrument response, and it is highlighted by the red curve in Figure 10.2 . In the particular example of Fig. 10.2 , the smoothing has been done by a boxcar average with 200 pixels width, as implemented in the SMOOTH IDL function. The response below 1.46 microns has been obtained as extrapolation of 1 degree polynomium fitted in the 1.46 and 1.49 microns range. The response above 1.83 has been set to constant.
- Repeat the above steps for all the valid extensions of telluric.fits and collect all the response curves.

11 Frequently Asked Questions

- **The error window fills the whole screen - how can I get to the `Continue`/`Stop` buttons?**

  Press the Alt key together with your left mouse button to move the window upwards and to the left. At the bottom the `Continue`/`Stop` buttons will be visible. This bug is known but could not yet be fixed.

- **I tried to `Open` (or `Configure`) an `Actor` while the workflow is running and now it does not react any more. What should I do?**

  This is a limitation of the underlying Kepler engine. The only way out is to kill the workflow externally. If you want to change anything while a workflow is running you first need to pause it.

- **Where are my intermediate pipeline products?** Intermediate pipeline products are stored in the directory `<TMP_PRODUCTS_DIR>` (defined on the workflow canvas, under Setup Directories) and organised further in directories by pipeline recipe.

- **Can I use different sets of bias frames to calibrate my flat frames and science data?** Yes. In fact this is what is currently implemented in the workflow(s). Each file in a DataSet has a purpose attached to it ([5]). It is this purpose that is used by the workflow to send the correct set of bias frames to the recipes for flat frame combination and science frame reduction, which may or may not be the same set of bias frames in each case.

- **Can I run `Reflex` from the command line?** Yes, use the command:

  ```
  esoreflex -n <workflow_path>/<workflow>.xml
  ```

  The -n option will set all the different options for Kepler and the workflows to avoid opening any GUI elements (including pipeline interactive windows).

  It is possible to specify workflow variables (those that appear in the workflow canvas) in the command line. For instance, the raw data directory can be set with this command:

  ```
  esoreflex -n -RAW_DATA_DIR <raw_data_path> \
               <workflow_path>/<workflow>.xml
  ```

  You can see all the command line options with the command `./esoreflex` -h.

  Note that this mode is not fully supported, and the user should be aware that the path to the workflow must be absolute and even if no GUI elements are shown, it still requires a connection to the window manager.

- **How can I add new actors to an existing workflow?** You can drag and drop the actors in the menu on the left of the `Reflex` canvas. Under `Eso-reflex -> Workflow` you may find all the actors relevant for pipeline workflows, with the exception of the recipe executer. This actor must be manually instantiated using `Tools -> Instantiate Component`. Fill in the "Class name" field with `org.eso.RecipeExecuter` and in the pop-up window choose the required recipe from the pull-down menu. To connect the ports of the actor, click on the source port, holding down the left mouse button, and release the mouse button over the destination port. Please consult the Reflex User Manual ([5]) for more information.

- **How can I broadcast a result to different subsequent actors?** If the output port is a multi-port (filled in white), then you may have several relations from the port. However, if the port is a single port (filled in black), then you may use the black diamond from the toolbar. Make a relation from the output port to the diamond. Then make relations from the input ports to the diamond. Please note that you cannot click to start a relation from the diamond itself. Please consult the Reflex User Manual ([5]) for more information.

- **How can I manually run the recipes executed by Reflex?** If a user wants to re-run a recipe on the command line he/she has to go to the appropriate reflex_book_keeping directory, which is generally reflex_book_keeping/<workflow>/<recipe_name>_<number> There, subdirectories exist with the time stamp of the recipe execution (e.g. 2013-01-25T12:33:53.926/). If the user wants to re-execute the most recent processing he/she should go to the `latest` directory and then execute the script `cmdline.sh`. Alternatively, to use a customized `esorex` command the user can execute

  ```
  ESOREX_CONFIG="INSTALL_DIR/etc/esorex.rc"
  PATH_TO/esorex --recipe-config=<recipe>.rc <recipe> data.sof
  ```

  where INSTALL_DIR is the directory where Reflex and the pipelines were installed.

  If a user wants to re-execute on the command line a recipe that used a specific raw frame, the way to find the proper data.sof in the bookkeeping directory is via `grep <raw_file> */data.sof`. Afterwards the procedure is the same as before.

  If a recipe is re-executed with the command explained above, the products will appear in the directory from which the recipe is called, and not in the reflex_tmp_products or reflex_end_products directory, and they will not be renamed. This does not happen if you use the `cmdline.sh` script.

- **If I enter "-" into an empty integer parameter of an interactive window it is automaticaly completed to "-1". Why?**

  The parameters are validated for correctness according to their type (e.g. string, integer, float). In the case of an integer or float parameter "-" alone is considered an invalid input and is therefore automatically completed to "-1". This is part of the validation of input done by the WxPython library.

- **Can I reuse the bookkeeping directory created by previous versions of the pipeline?**

  In general no. In principle, it could be reused if no major changes were made to the pipeline. However there are situations in which a previously created bookkeeping directory will cause problems due to pipeline versions incompatibility. This is especially true if the parameters of the pipeline recipes have changed. In that case, please remove the bookkeeping directory completely.

12 Troubleshooting

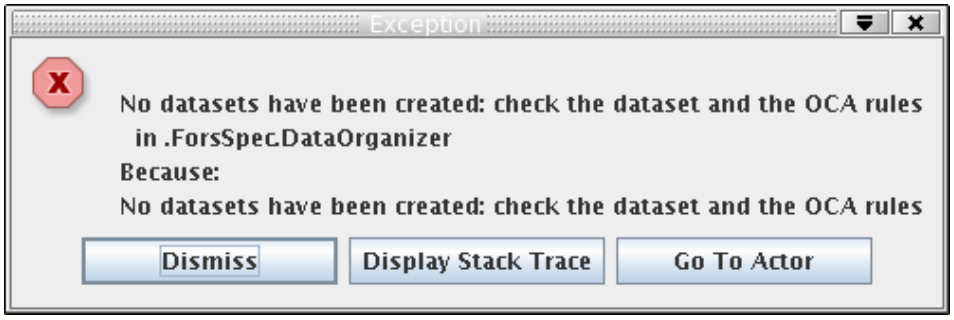


Figure 12.1: *TheDataOrganizer interactive window reports an error ":No DataSets have been created, check the data set and the OCA rules.".*

1. **I downloaded the data from the ESO archive, put them into a new directory, tried to run `Reflex` on them, but**

   (a) **it crashs**

   This may happen if one of the files was downloaded only partially (check for a file with the extension `fits.Z.part`. You will have to download that file again in order to have an uncorrrupted file (and remove the partial one).

   (b) **it fails with error message ":No DataSets have been created, check the data set and the OCA rules."(see Figure 12.1.)**

   This error may be due to the fact that the data provided by the ESO archive are compressed (`<filename>.fits.Z`). Please remember to uncompress the data before executing `Reflex`.

   (c) **all DataSets are greyed out in the DataSets interactive window.**

   The ESO archive used with CalSelector does not always supply all static calibration files. As a consequence some/all DataSets are greyed out because they were missing such required data.

   Missing static calibration should be found by `reflex` in `<install_directory>/calib/<pipeline_version>/cal`.

2. **The "Select DataSets" window displays my DataSets, but some/all of them are greyed out. What is going on?**

   If a DataSet in the "Select DataSets" window is greyed out, then it means that the DataSet that was constructed is missing some key calibration(s) (i.e. the DataSet is incomplete). To find out what calibration(s) are missing from a greyed out DataSet, click on the DataSet in question to highlight it in blue, and then click on the button `Inspect Highlighted`. The "Select Frames" window that appears will report the category of the calibration products that are missing (e.g. `MASTER_BIAS`). From this the user has then to determine the missing raw data (in this case bias frames). If static calibrations are missing the mechanism unfortunately does not work, but should be found by `reflex` in `<install_directory>/calib/<pipeline_version>/cal`

3. **The plots in the interactive windows does not allow me to properly inspect the products; how can I change or measure what it is plotted?**

   The plots in the interactive window are meant to provide a general visualization of the product. However, it is possible to inspect each file (input or output) with other visualization tools, or process them via custom scripts to evalueta the quality.

   At the bottom rigth corner of each interactive window, the list of inputs/outputs files is given. Select with the mouse the file you would like to inspect and press Ctrl-C to copy its full path name.

   It is also possible to change the general visualization tool in the Global Parameter section of the workflow, by editing the `FITS_VIEWER` variable.

Acknowledgements

The Reflex team in alphabetical order consists of Pascal Ballester, Daniel Bramich, Vincenzo Forchì, Wolfram Freudling, César Enrique García, Maurice Klein Gebbinck, Andrea Modigliani, Sabine Moehler, Mark Neeser & Martino Romaniello.

[1] Forchì V., 2012, Reflex User Manual, VLT-MAN-ESO-19000-5037, Issue 0.7

[2] Davies R., Berbel, A., Wiezorrek E., Ott T., Schreiber N. 2012, KMOS Data Flow: Reconstructing Data Cubes in One Step, Max-Planck Institut für extraterrestrische Physik. 7

[3] Davies R., Berbel, A., Wiezorrek E. 2014, SPARK INstructional Guide ofr KMOS data, VLT-MAN-KMO-146611-009, issue 1.5. 44

[4] Cirasuolo M., Sharples R., 2013, KMOS User Manual, VLT-MAN-KMO-146603-001, Issue 5.0 7

[5] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, *http://www.eso.org/reflex/*, 2012. VLT-MAN-ESO-19000-5037. 25, 55, 56