VERY LARGE TELESCOPE

Molecfit Reflex Tutorial

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

Issue 3.0.3
Date 2020-09-05
Change record

<table>
<thead>
<tr>
<th>Issue/Rev.</th>
<th>Date</th>
<th>Section affected</th>
<th>Reason/Initiation/Documents/Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>03/06/2020</td>
<td>All</td>
<td>A new Molecfit Pipeline has been released</td>
</tr>
<tr>
<td>4.0</td>
<td>01/08/2020</td>
<td>1, 3, 5-7</td>
<td>New workflow version; support for XSHOOTER and UVES.</td>
</tr>
</tbody>
</table>
Contents

1 Introduction to Esoreflex

2 Software Installation

2.1 Installing Reflex workflows via macports

2.2 Installing Reflex workflows via rpm/yum/dnf

2.3 Installing Reflex workflows via install_esoreflex

2.4 Demo Data

3 Quick Start: Reducing The Demo Data

4 About the main esoreflex canvas

4.1 Saving And Loading Workflows

4.2 Buttons

4.3 Workflow States

5 The Molecfit Workflow

5.1 Workflow Canvas Parameters

5.2 MOLECFIT-specific workflow parameters

5.3 Workflow Actors

5.3.1 Simple Actors

5.3.2 Lazy Mode

6 Reducing your own data

6.1 The esoreflex command

6.2 Launching the workflow

6.3 Workflow Steps

6.3.1 Data Organisation And Selection

6.3.2 DataSetChooser

6.3.3 The workflow data-reduction cascade

6.3.4 The ProductExplorer

6.4 Description of the final Molecfit data
6.4.1 Temporary and End products for demo datasets ........................................... 28

7 Optimising Your Results Through Workflow Interaction ................................. 29
  7.1 General characteristics of the interactive windows .................................. 29
  7.2 Computation of atmospheric model: running molecfit_model on selected spectra ...... 31
    7.2.1 Tips for an efficient fit. ................................................................. 32
  7.3 Computation of atmospheric transmission: running molecfit_calctrans ......... 34
  7.4 Correction for atmospheric transmission: running molecfit_correct ............ 35

8 Frequently Asked Questions ........................................................................... 37

9 Troubleshooting ......................................................................................... 40
1 Introduction to Esoreflex

This document is a tutorial designed to enable the user to reduce his/her data with the ESO pipeline run under an user-friendly environment, called Esoreflex, concentrating on high-level issues such as data reduction quality and signal-to-noise (S/N) optimisation.

Esoreflex 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.

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 telluric correction of spectroscopic observations, by using the the Molecfit esoreflex molecfit.xml workflow. Molecfit models the atmosphere by fitting an observed spectrum and then generate the transmission correction for a given science target target. The user can decide whether to use the target spectrum to generate the atmospheric model, or the spectrum of a telluric standard star observed the same night.

The current version supports the telluric correction of XSHOOTER (all instrument modes except IFU or UVB arm) and UVES (REDL and REDU data configuration) data as produced by the corresponding instrument pipelines. Support for XSHOOTER is also extended to reduced data present in the ESO archive. Expert users can however adapt the workflow to support other instruments.

The quick start section (see Section 3) describes the minimum effort to get started, and it makes up only few pages in this tutorial.

\(^1\)http://www.eso.org/sci/archive/calselectorInfo.html
2 Software Installation

Esoreflex 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 macOS 10.11 to 10.14, while the rpm/yum repositories support Fedora 28 to 31, CentOS 7, Scientific Linux 7. For any other operating system it is recommended to use the install_esoreflex script.

The installation from package repository requires administrative privileges (typically granted via sudo), as it installs files in system-wide directories under the control of the package manager. If you want a local installation, or you do not have sudo privileges, or if you want to manage different installations on different directories, then use the install_esoreflex script. Note that the script installation requires that your system fulfill several software prerequisites, which might also need sudo privileges.

Reflex 2.10 needs java JDK 11 to be installed.

Please note that in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the $HOME/KeplerData, $HOME/.kepler directories if present, to prevent possible aborts (i.e. a hard crash) of the esoreflex process.

2.1 Installing Reflex workflows via macports

This method is supported for the macOS operating system. It is assumed that macports (http://www.macports.org) is installed. Please read the full documentation at http://www.eso.org/sci/software/pipelines/installation/macports.html.

2.2 Installing Reflex workflows via rpm/yum/dnf

This method is supported for Fedora 26 to 29, CentOS 7, Scientific Linux 7 operating systems, and requires sudo rights. To install, please follow these steps

1. Configure the ESO repository (This step is only necessary if the ESO repository has not already been previously configured).
   - If you are running Fedora 26 or newer, run the following commands:
     
     sudo dnf install dnf-plugins-core  
   - If you are running CentOS 7, run the following commands:
     
     sudo yum install yum-utils ca-certificates yum-conf-repos  
     sudo yum install epel-release  
• If you are running SL 7, run the following commands:
  
  sudo yum install yum-utils ca-certificates yum-conf-repos  
  sudo yum install yum-conf-epel  
  pipelines/repositories/stable/sl/esorepo.repo

2. Install the pipelines

• The list of available top level packages for different instruments is given by:
  
  sudo dnf list esopipe-\/*/all # (Fedora 26 or newer)  
  sudo yum list esopipe-\/*/all # (CentOS 7, SL 7)

• To install an individual pipeline use the following (This example is for X-Shooter. Adjust the package name to the instrument you require):
  
  sudo dnf install esopipe-xshoo-all # (Fedora 26 or newer)  
  sudo yum install esopipe-xshoo-all # (CentOS 7, SL 7)

• To install all pipelines use:
  
  sudo dnf install esopipe-\/*/all # (Fedora 26 or newer)  
  sudo yum install esopipe-\/*/all # (CentOS 7, SL 7)

For further information, please read the full documentation at

2.3 Installing Reflex workflows via install_esoreflex

This method is recommended for operating systems other than what indicated above, or if the user has no sudo rights. Software dependencies are not fulfilled by the installation script, therefore the user has to install all the prerequisites before running the installation script.

The software pre-requisites for Reflex 2.11 may be found at:
http://www.eso.org/sci/software/pipelines/reflex_workflows

To install the Reflex 2.11 software and demo data, please follow these instructions:

1. From any directory, download the installation script:


2. Make the installation script executable:

   chmod u+x install_esoreflex

3. Execute the installation script:
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. Follow all the script instructions; you will be asked whether to use your Internet connection (recommended: yes), the pipelines and demo-datasets to install (note that the installation will remove all previously installed pipelines that are found in the same installation directory).

5. 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.

### 2.4 Demo Data

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

In the demo dataset folder, the raw input sets of data defines the following datasets:

- **XSHOOTER_NIR** : Near Infrared.
- **XSHOOTER_VIS** : Visible.
- **TELLURIC_VIS.fits** : Visible. This is a telluric standard star, it can be considered a stand-alone dataset to be corrected, or a reference spectrum to use for the correction of XSHOOTER_VIS.
- **UVES_REDL** : Visible
3 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 Molecfit demo data set supplied with the esoreflex 2.11 release. By following these steps, the user should have enough information to perform a reduction of his/her own data without any further reading:

1. First, type:

   esoreflex -l

   If the esoreflex executable is not in your path, then you have to provide the command with the executable full path <install_dir>/bin/esoreflex -l. For convenience, we will drop the reference to <install_dir>. A list with the available esoreflex workflows will appear, showing the workflow names and their full path.

2. Open the molecfit by typing:

   esoreflex molecfit&

   Alternatively, you can type only the command esoreflex the empty canvas will appear (Figure 3.1) and you can select the workflow to open by clicking on File -> Open File. Note that the loaded workflow will appear in a new window. The molecfit workflow is shown in Figure 3.2.

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. Change directories set-up. Under “Setup Directories” in the workflow canvas there are seven parameters that specify important directories (green dots).

   By default, the ROOT_DATA_DIR, which specifies the working directory within which the other directories are organised, is set to your $HOME/reflex_data directory. All the temporary and final products of the reduction will be organized under sub-directories of ROOT_DATA_DIR, therefore make sure this parameter points to a location where there is enough disk space. To change ROOT_DATA_DIR, double click on it 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.

   Changing the value of RAW_DATA_DIR is the only necessary modification if you want to process data other than the demo data.

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 `esoreflex` 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 3) that lists the datasets along with the values of a selection of useful header keywords\(^2\). 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. A full description of the options offered by the **Data Set Chooser** will be presented in Section 6.3.2.

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. Once the reduction of all datasets has finished, a pop-up window called **Product Explorer** will appear, showing the datasets which have been 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 6.2 shows the Product Explorer window. A full description of the **Product Explorer** will be presented in Section 6.3.4.

10. After the workflow has finished, all the products from all the datasets can be found in a directory under `END_PRODUCTS_DIR` named after 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 Molecfit workflow that merit a look at the rest of this tutorial.

\(^2\)The keywords listed can be changed by double clicking on the **DataOrganiser Actor** and editing the list of keywords in the second line of the pop-up window. Alternatively, instead of double-clicking, you can press the right mouse button on the **DataOrganiser Actor** and select **Configure Actor** to visualize the pop-up window.
Note 2: When the workflow starts, the user will be warned that some recipe parameters have been re-initialized according to the chosen instrument. For the demo dataset, the user can continue with the processing by simply ignoring the warning. The parameter initialization will be explained in Section 5.2.

Note 3: 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 7; press the “Continue” button to proceed with the workflow execution.

Figure 3.2: Molecfit workflow general layout.
Figure 3.3: The “Select Datasets” pop-up window.
4 About the main esoreflex canvas

4.1 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 esoreflex 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.

4.2 Buttons

At the top of the esoreflex 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.

4.3 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.
5 The Molecfit Workflow

The Molecfit 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.

5.1 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; [3]).

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 CalSelector associations in the Data Organiser properties and make sure that the input data directory contains the XML file downloaded with the CalSelector archive request (note that this does not work for all instrument workflows).

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 5.3.2), reusing the previous pipeline recipe outputs if 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 re-reduce 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 failures 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`, `Triggered`, and `Disabled`. `Enabled` opens the `ProductExplorer` GUI at the end of the reduction of each individual dataset. `Triggered` (default and recommended) opens the `ProductExplorer` GUI when all the selected datasets have been reduced. `Disabled` does not display the `ProductExplorer` GUI.

### 5.2 MOLECFIT-specific workflow parameters

The general Molecfit workflow canvas contains several parameters that are specific to Molecfit. They allow the user to select the best initial configuration for the desired instrument and the data reduction strategy. These parameters need to be set before starting the workflow.

- **INSTRUMENT.** Allows to select the instrument the data to be corrected are taken. Only data for the selected instrument will be considered for processing. Available options are: XSHOOTER, UVES. Support for additional instruments will be provided in future releases.

- **ParameterInitialization.** If set to true (lower case) the workflow sets some recipe parameter to default values according to the instrument the data are taken. It is recommended to set it to true, unless the user intends to use values that have been optimized from previous executions (and stored, for example, by setting “Use the parameters above as initial values in subsequent executions of this recipe”). If set to false, the values of the parameter currently present in the workflow will be used. Note that, if the set-up is not suited for the current dataset, some recipes might fail, causing the workflow to crash. If this happens, the workflow reduction needs to be restarted with ParameterInitialization = true. In future releases, it will be possible to change recipe parameters from interactive windows also after recipe crashes, avoiding the need to restart the reduction.

    If this option is true, then the workflow initializes the following `molecfit_model` recipe parameters:

    - `CONTINUUM_N`. It specifies the polynomial degree for fitting the spectrum continuum.
    - `LIST_MOLEC`. It specifies the list of molecules to be used in the fit.
    - `REL_COL`. It specifies the relative column densities of the molecules.
    - `WLG_TO_MUCRON`. It specifies the conversion factor between the wavelength units and microns.
    - `FIT_MOLEC`. It specifies whether a given molecule is fitted or computed.
    - `FIT_TELESCOPE_BACKGROUND`. It specifies whether the telescope background has to be fitted or not.

    For more information on these parameters, please consult Section 7.2.1 or the `molecfit` reference manual ([2]).

    When starting the workflow, the user will be warned whether ParameterInitialization is set to true or false. To enable/disable the warning, double click on the actor "Check Parameter Initialization" (third actor along the workflow chain) and set the variable “Show warnings on parameter initialization?” to Yes or No to enable or disable the warning, respectively.
• **Use the science spectrum for fitting the atmosphere.** If true, then the science frame is used to fit the atmospheric parameters. If false, then the telluric star is used to fit the atmospheric parameters. If the telluric star is not present, then the science exposure is used with no warnings. Default: false.

Typically, the best corrections are obtained if the atmospheric model is computed directly from the science spectrum to correct. However, for faint targets (e.g., $S/N < 50$ per Å) or for targets that shows many intrinsic spectral features in the wavelength regions used in the fit, it might be worth using a telluric standard star for this purpose.

The workflow automatically selects the spectrum of a telluric standard if present in the input directory and if compatible with the spectrum to correct (e.g., must be observed with the same instrument configuration and on the same night as the science spectrum to correct).

• **GlobalEnableKernel**: Enable/Disable the use of calibration files describing the instrument line spread function (kernel). The current molecfit workflow release do not include any kernel. If no kernel is provided, this option has no effects in the process. Default: true.

### 5.3 Workflow Actors

#### 5.3.1 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:

- **DataOrganiser**

- **DataSetChooser** (inside a composite actor).

- **FitsRouter** actor Redirects files according to their categories.

- **ProductRenamer** actor.

- **ProductExplorer** 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).
5.3.2 Lazy Mode

By default, all RecipeExecuter 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 re-reduction where it is not necessary.

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

The re-reduction of data at each execution may sometimes be desirable. To force a re-reduction of data for any single RecipeExecuter actor in the workflow, right-click the actor, select Configure Actor, and uncheck the Lazy mode parameter tick-box in the “Edit parameters” window that is displayed. For many workflows the RecipeExecuter actors are actually found inside the composite actors in the top level workflow. To access such embedded RecipeExecuter actors you will first need to open the sub-workflow by right-clicking on the composite actor and then selecting Open Actor.

To force the re-reduction of all data in a workflow (i.e. to disable Lazy mode for the whole workflow), you must uncheck the Lazy mode for every single RecipeExecuter actor in the entire workflow. It is also possible to change the name of the bookkeeping directory, instead of modifying any of the Lazy mode parameters. This will also force a re-reduction of the given dataset(s). A new reduction will start (with the lazy mode still enabled), but the results of previous reduction will not be reused. Alternatively, if there is no need to keep any of the previously reduced data, one can simply set the EraseDirs parameter under the “Global Parameters” area of the workflow canvas to true. This will then remove all previous results that are stored in the bookkeeping, temporary, and log directories before processing the input data, in effect, starting a new clean data reduction and re-processing every input dataset. Note: The option EraseDirs = true does not work in esoreflex version 2.9.x and makes the workflow to crash.
6 Reducing your own data

In this section we describe how to reduce your own data set.

First, we suggest the reader to familiarize with the workflow by reducing the demo dataset first (Section 3), but it is not a requirement.

6.1 The esoreflex command

We list here some options associated to the esoreflex command. We recommend to try them to familiarize with the system. In the following, we assume the esoreflex executable is in your path; if not you have to provide the full path <install_dir>/bin/esoreflex

To see the available options of the esoreflex command type:

esoreflex -h

The output is the following.

-h | -help print this help message and exit.
-v | -version show installed Reflex version and pipelines and exit.
-l | -list-workflows list available installed workflows and from ~/KeplerData/workflows.
-n | -non-interactive enable non-interactive features.
-e | -explore run only the Product Explorer in this workflow
-p <workflow> | -list-parameters <workflow> lists the available parameters for the given workflow.
-config <file> allows to specify a custom esoreflex.rc configuration file.
-create-config <file> if <file> is TRUE then a new configuration file is created in ~/.esoreflex/esoreflex.rc. Alternatively a configuration file name can be given to write to. Any existing file is backed up to a file with a ’.bak’ extension, or ’.bakN’ where N is an integer.
-debug prints the environment and actual Reflex launch command used.

6.2 Launching the workflow

We list here the recommended way to reduce your own datasets. Steps 1 and 2 are optional and one can start from step 3.

1. Type: esoreflex -n <parameters> molecfit to launch the workflow non interactively and reduce all the datasets with default parameters.
<parameters> allows you to specify the workflow parameters, such as the location of your raw data and the final destination of the products.

For example, type (in a single command line):

esoreflex -n
   -RAW_DATA_DIR /home/user/my_raw_data
   -ROOT_DATA_DIR /home/user/my_reduction
   -END_PRODUCTS_DIR $ROOT_DATA_DIR/reflex_end_products
molecfit

to reduce the complete datasets that are present in the directory /home/user/my_raw_data and that were not reduced before. Final products will be saved in /home/user/my_reduction/reflex_end_products, while book keeping, temporary products, and logs will be saved in sub-directories of /home/user/my_reduction/. If the reduction of a dataset fails, the reduction continues to the next dataset. It can take some time, depending on the number of datasets present in the input directory. For a full list of workflow parameters type esoreflex -p molecfit. Note that this command lists only the parameters, but does not launch the workflow.

Once the reduction is completed, one can proceed with optimizing the results with the next steps.

2. Type:

esoreflex -e molecfit

to launch the Product Explorer. The Product Explorer allows you to inspect the data products already reduced by the molecfit esoreflex workflow. Only products associated with the workflow default bookkeeping database are shown. To visualize products associated to given bookkeeping database, pass the full path via the BOOKKEEPING_DB parameter:

esoreflex -e BOOKKEEPING_DB <database_path> molecfit

to point the product explorer to a given <database_path>, e.g., /home/username/reflex/reflect_bookkeeping/test.db

The Product Explorer allows you to inspect the products while the reduction is running. Press the button [Refresh] to update the content of the Product Explorer. This step can be launched in parallel to step 1.

A full description of the Product Explorer will be given in Section 6.3.4

3. Type:

esoreflex molecfit &

to launch the molecfit esoreflex workflow. The molecfit workflow window will appear (Fig. 3.2). Please configure the set-up directories ROOT_DATA_DIR, RAW_DATA_DIR, and other workflow parameters as needed. Just double-click on them, edit the content, and press [OK]. Remember to specify the same <database_path> as for the Product Explorer, if it has been opened at step #2, to synchronize the two processes.

4. (Recommended, but not mandatory) On the main esoreflex menu set Tools –> Animate at Runtime to 1 in order to highlight in red active actors during execution.

5. Press the button [ ] to start the workflow. First, the workflow will highlight and execute the Initialise actor, which among other things will clear any previous reductions if required by the user (see Section 5.1).
Secondly, if set, the workflow will open the Product Explorer, allowing the user to inspect previously reduced datasets (see Section 6.3.4 for how to configure this option).

6.3 Workflow Steps

6.3.1 Data Organisation And Selection

The DataOrganiser (DO) is the first crucial component of a Reflex workflow. The DO takes as input RAW_DATA_DIR and CALIB_DATA_DIR and it detects, classifies, and organises the files in these directories and any subdirectories. The output of the DO is a list of “DataSets”. A DataSet is a special Set of Files (SoF). A DataSet contains one or several science (or calibration) files that should be processed together, and all files needed to process these data. This includes any calibration files, and in turn files that are needed to process these calibrations. Note that different DataSets might overlap, i.e. some files might be included in more than one DataSet (e.g., common calibration files).

A DataSet lists three different pieces of information for each of its files, namely 1) the file name (including the path), 2) the file category, and 3) a string that is called the “purpose” of the file. The DO uses the OCA rules to find the files to include in a DataSet, as well as their categories and purposes. The file category identifies different types of files, and it is derived by information in the header of the file itself. A category could for example be RAW_CALIBRATION_1, RAW_CALIBRATION_2 or RAW_SCIENCE, depending on the instrument. The purpose of a file identifies the reason why a file is included in a DataSet. The syntax is action_1/action_2/action_3/ ... /action_n, where each action_i describes an intended processing step for this file (for example, creation of a MASTER_CALIBRATION_1 or a MASTER_CALIBRATION_2). The actions are defined in the OCA rules and contain the recipe together with all file categories required to execute it (and predicted products in case of calibration data). For example, a workflow might include two actions action_1 and action_2. The former creates MASTER_ CALIBRATION_1 from RAW_CALIBRATION_1, and the later creates a MASTER_ CALIBRATION_2 from RAW_CALIBRATION_2. The action_2 action needs RAW_CALIBRATION_2 frames and the MASTER_ CALIBRATION_1 as input. In this case, these RAW_CALIBRATION_1 files will have the purpose action_1/action_2. The same DataSet might also include RAW_CALIBRATION_1 with a different purpose; irrespective of their purpose the file category for all these biases will be RAW_CALIBRATION_1.

The DataSets created via the DataOrganiser will be displayed in the DataSet Chooser. Here the users have the possibility to inspect the various datasets and decide which one to reduce. By default, DataSets that have not been reduced before are highlighted for reduction. Click either [Continue] in order to continue with the workflow reduction, or [Stop] in order to stop the workflow. A full description of the DataSet Chooser is presented in Section 6.3.2.

Once the [Continue] is pressed, the workflow starts to reduce the first selected DataSet. Files are broadcasted according to their purpose to the relevant actors for processing.

The categories and purposes of raw files are set by the DO, whereas the categories and purpose of products generated by recipes are set by the RecipeExecutor. The file categories are used by the FitsRouter

3OCA stands for OrganisationClassificationAssociation and refers to rules, which allow to classify the raw data according to the contents of the header keywords, organise them in appropriate groups for processing, and associate the required calibration data for processing. They can be found in the directory <install_dir>/share/esopipes/<pipeline-version>/reflex/, carrying the extension.oca
to send files to particular processing steps or branches of the workflow (see below). The purpose is used by the SofSplitter and SofAccumulator to generate input SoFs for the RecipeExecuter. The SofSplitter and SofAccumulator accept several SoFs as simultaneous input. The SofAccumulator creates a single output SoF from the inputs, whereas the SofSplitter creates a separate output SoF for each purpose.

### 6.3.2 DataSetChooser

The DataSetChooser displays the DataSets available in the “Select Data Sets” window, activating vertical and horizontal scroll bars if necessary (Fig. 3).

Some properties of the DataSets are displayed: the name, the number of files, a flag indicating if it has been successfully reduced (a green OK), if the reduction attempts have failed or were aborted (a red FAILED), or if it is a new dataset (a black "). The column "Descriptions" lists user-provided descriptions (see below), other columns indicate the instrument set-up and a link to the night log.

Sometimes you will want to reduce a subset of these DataSets rather than all DataSets, and for this you may individually select (or de-select) DataSets for processing using the tick boxes in the first column, and the buttons Deselect All and Select Complete at the bottom, or configure the “Filter” field at the bottom left. Available filter options are: "New" (datasets not previously reduced will be selected), "Reduced" (datasets previously reduced will be selected), "All" (all datasets will be selected), and "Failed" (dataset with a failed or aborted reduction will be selected).

You may also highlight a single DataSet in blue by clicking on the relevant line. If you subsequently click on Inspect Highlighted, then a “Select Frames” window will appear that lists the set of files that make up the highlighted DataSet including the full filename, the file category (derived from the FITS header), and a selection tick box in the right column. The tick boxes allow you to edit the set of files in the DataSet which is useful if it is known that a certain calibration frame is of poor quality (e.g: a poor raw flat-field frame). The list of files in the DataSet may also be saved to disk as an ASCII file by clicking on Save As and using the file browser that appears.

By clicking on the line corresponding to a particular file in the “Select Frames” window, the file will be highlighted in blue, and the file FITS header will be displayed in the text box on the right, allowing a quick inspection of useful header keywords. If you then click on Inspect, the workflow will open the file in the selected FITS viewer application defined by the workflow parameter FITS_VIEWER.

To exit from the “Select Frames” window, click Continue.

To add a description of the reduction, press the button associated with the field "Add description to the current execution of the workflow” at the bottom right of the Select Dataset Window; a pop up window will appear. Enter the desired description (e.g: "My first reduction attempt") and then press OK. In this way, all the datasets reduced in this execution, will be flagged with the input description. Description flags can be visualized in the SelectFrames window and in the ProductExplorer, and they can be used to identify different reduction strategies.

To exit from the “Select DataSets” window, click either Continue in order to continue with the workflow reduction, or Stop in order to stop the workflow.

---

4keep the mouse pointer on the file name to visualize the full path name.
6.3.3 The workflow data-reduction cascade

The present Molecfit 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:

- **Model the atmosphere**: it executes the recipe `molecfit_model`. The recipe fits an atmospheric model to the input spectrum (either a telluric standard or the science spectrum to be corrected) to determine column densities of several molecules. In the process, several parameters that depend on the instrument properties (such as telescope background, spectral resolution, and wavelength calibration accuracy) are taken into account. Note: the wavelength correction determined in molecfit_model is used only for the model, and not used to refine the wavelength calibration of the final science spectrum. Future version of the workflow will allow to select the desired options.

  We refer the user to the Molecfit User Manual [2] for further information. The products of this recipe are `ATMOS_PARM` (a table with atmospheric parameters such as pressure, temperature, and humidity), `BEST_FIT_PARM` (a table containing the best fit atmospheric and instrumental parameters), and `BEST_FIT_MODEL` (the best fit model to the data).

  This is an interactive actor, meaning that the results are displayed on an interactive window for inspection. The user has the possibility modify the recipe parameters and re-run this step. The interactive features are described in Section 7.2.

- **Compute telluric correction**: it executes the recipe `molecfit_calctrans`. It uses the atmospheric model computed by `molecfit_model` and one reduced scientific observation to compute the atmospheric transmission over all the wavelength range of the scientific data. It is possible to use `molecfit_model` on a standard star and use `molecfit_calctrans` on a science target: in this case `molecfit_calctrans` will take the difference of airmass into account. If a kernel library is used, `molecfit_calctrans` can be configured to use a different line spread function than the one used by `molecfit_model`. We refer the user to the Molecfit User Manual [2] for further information.

  This is an interactive actor, meaning that the results are displayed on an interactive window for inspection. The user has the possibility modify the recipe parameters and re-run this step. The interactive features are described in Section 7.3.

- **Correct data**: it executes the recipe `molecfit_correct` It applies the telluric correction computed in Atmospheric transmission to the science data.

  This is an interactive actor, meaning that the results are displayed on an interactive window for inspection. The user has the possibility modify the recipe parameters and re-run this step. The interactive features are described in Section 7.4.

We refer the user to the Molecfit pipeline manual for a complete description of the recipes and their parameters.
Figure 6.1: The “Selected Frame” pop-up window, obtained after pressing Inspect highlighted in the Select Datasets window (Fig. 3).

Some of the above actors trigger an interactive window, allowing the user to inspect 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 7.

As noted in the Quick Start Section 3, 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 6.4. The intermediate pipeline calibration products can be found in subdirectories of the TMP_PRODUCT_DIR.

Figure 6.2: The “Product Explorer” pop-up window showing final results of the workflow.
6.3.4 The ProductExplorer

The ProductExplorer is an interactive component in the esoreflex workflow whose main purpose is to list the final products with the associated reduction tree for each dataset and for each reduction attempt (see Fig. 6.2).

Configuring the ProductExplorer

You can configure the ProductExplorer GUI to appear after or before the data reduction. In the latter case you can inspect products as reduction goes on.

1. To display the ProductExplorer GUI at the end of the data reduction:

   - Click on the global parameter “ProductExplorerMode” before starting the data reduction. A configuration window will appear allowing you to set the execution mode of the Product Explorer. Valid options are:
     - "Triggered" (default). This option opens the ProductExplorer GUI when all the selected datasets have been reduced.
     - "Enabled". This option opens the ProductExplorer GUI at the end of the reduction of each individual dataset.
     - “Disable”. This option does not display the ProductExplorer GUI.

   - Press the button to start the workflow.

2. To display the ProductExplorer GUI “before” starting the data reduction:

   - Double click on the composite Actor "Inspect previously reduced data". A configuration window will appear. Set to "Yes" the field "Inspect previously reduced data (Yes/No)". Modify the field "Continue reduction after having inspected the previously reduced data? (Continue/Stop/Ask)". "Continue" will continue the workflow and trigger the DataOrganizer. "Stop" will stop the workflow; "Ask" will prompt another window deferring the decision whether continuing or not the reduction after having closed the Product Explorer.

   - Press the button to start the workflow. Now the ProductExplorer GUI will appear before starting the data organization and reduction.

Exploring the data reduction products

The left window of the ProductExplorer GUI shows the executions for all the datasets (see Fig. 6.2). Once you click on a dataset, you get the list of reduction attempts. Green and red flags identify successful or unsuccessful reductions. Each reduction is linked to the “Description” tag assigned in the “Select Dataset” window.

1. To identify the desired reduction run via the “Description” tag, proceed as follows:

   - Click on the symbol at the left of the dataset name. The full list of reduction attempts for that dataset will be listed. The column Exec indicates if the reduction was successful (green flag: "OK") or not (red flag: “Failed”).
• Click on the entries in the field "Description" to visualize the description you have entered associated to that dataset on the Select Dataset window when reducing the data.

• Identify the desired reduction run. All the products are listed in the central window, and they are organized following the data reduction cascade.

You can narrow down the range of datasets to search by configuring the field "Show" at the top-left side of the ProductExplorer (options are: "All", "Successful", "Unsuccessful"), and specifying the time range (Last, all, From-to).

2. To inspect the desired file, proceed as follows:

• Navigate through the data reduction cascade in the ProductExplorer by clicking on the files.

• Select the file to be inspected and click with the mouse right-hand button. The available options are:

  – Options available always:
    * Copy full path. It copies the full name of the file onto the clipboard. Shift+Ctrl+v to past it into a terminal.
    * Inspect Generic. It opens the file with the fits viewer selected in the main workflow canvas.
    * Inspect with. It opens the file with an executable that can be specified (you have to provide the full path to the executable).

  – Options available for files in the TMP_PRODUCTS_DIR directory only:
    * command line. Copy of the environment configuration and recipe call used to generate that file.
    * Xterm. It opens an Xterm at the directory containing the file.

  – Options available for products associated to interactive windows only:
    * Display pipeline results. It opens the interactive windows associated to the recipe call that generated the file. Note that this is for visualization purposes only; the recipe parameters cannot be changed and the recipe cannot be re-run from this window.

6.4 Description of the final Molecfit data

A number of intermediate pipeline products from the tutorial data set can be found in subdirectories of the REFLEX_TMP_PRODUCTS. We refer to the molecfit pipeline manual for a description of the products of each recipe.

The final products of the reduction pipeline can be found in the directory REFLEX_END_PRODUCTS that is defined in the “Setup Directories” section at the top of the workflow.

Workflow logs including the esorex commands executed can be found in the REFLEX_LOGS.

The end products from the tutorial data sets are summarized below.
6.4.1 Temporary and End products for demo datasets

- reflex_end_products
  - R71_A_SCIENCE_TELLURIC_CORR.fit. Obtained from the XSHOOTER_NIR dataset.
  - DLA1_PA3_long_SCIENCE_TELLURIC_CORR.fits. Obtained from the XSHOOTER_VIS dataset.
  - Telluric_Standard_SCIENCE_TELLURIC_CORR.fits. Obtained from TELLURIC_VIS dataset.
  - hd37020_dic2_SCIENCE_TELLURIC_CORR.fits. Obtained from UVES_REDL dataset.

Note: the products category is SCIENCE_TELLURIC_CORR, as defined by the header keyword: HIERARCH ESO PRO CATG. Their format reflect those of the input science files.

- reflex_tmp_products Note: the files in these lists are identified by their category as defined by the header keyword: HIERARCH ESO PRO CATG
  - molecfit_model
    * BEST_FIT_MODEL. Full outcome of molecfit_model, including the best fit to the input data and wavelength correction.
    * PIXEL_EXCLUDE. Table listing the pixels excluded in the fit.
    * BEST_FIT_PARAMETERS. Best fitting values of all the model parameters.
    * GDAS_BEFORE. Atmospheric profile (pressure, temperature and relative humidity as function of height) from GDAS database closest in time and precedent to the observation.
    * GDAS_AFTER. Atmospheric profile (pressure, temperature and relative humidity as function of height) from GDAS database closest in time and after to the observation.
    * GDAS. Atmospheric GDAS profile used in the fit, obtained as linear combination between GDAS_BEFORE and GDAS_AFTER.
    * ATM_PROFILE_STANDARD. Reference atmospheric parameter from database.
    * ATM_PROFILE_COMBINED. Reference atmospheric parameter from database combined with GDAS profile
    * ATM_PARAMETERS. Fitted atmospheric parameters (pressure, temperature, column density of molecules) as function of height.
    * MODEL_MOLECULES. Definition of the molecules and their initial fit parameters that are used in the model, as defined by the recipe parameters REL_COL, FIT_MOLEC, and LIST_MOLEC
    * WAVE_INCLUDE. Definition of the Wavelength regions used in the fit.

- molecfit_caltrans
  * LBLRTM_RESULTS. Details about the call of the linear transfer code.
  * TELLURIC_DATA. Full outcome of molecfit_caltrans.
  * TELLURIC_CORR. Atmospheric transmission; this is the telluric correction to apply to the input data.

- molecfit_correct
  * SCIENCE_TELLURIC_CORR. Corrected spectrum, same format as input spectrum.
  * SPECTRUM_TELLURIC_CORR. Corrected spectrum, binary image format.
7 Optimising Your Results Through Workflow Interaction

In this Section, we use the information from Section 3 along with one Molecfit demo data supplied with the installation 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 XSHOOTER_NIR dataset (object name: R71); information related to other second datasets 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 3, 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 3.

As discussed previously, the first operation the Molecfit workflow does is to group the input data (science frames, telluric stars, 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 Molecfit workflow triggers the recipes according to the data reduction cascade outlined in Section 6.3.3.

The recipes executed by the workflow trigger an interactive window, allowing the user to inspected the products and, eventually, to re-run the recipe with modified parameters. Section 7.1 describes the common properties of the interactive windows. Sections 7.2-7.4 deal with the individual interactive windows.

To disable the interactive windows, set the workflow parameter `GlobalPlotInteractivity` to false.

Note, if a recipe crashes, the corresponding interactive window also crashes. Therefore, one has to re-start the workflow. This behavior will be modified in future release: the interactive window will be displayed anyway (with no plots) allowing the user to change the recipe parameters and re-execute the step.

7.1 General characteristics of the interactive windows

For each of the 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.

The right-hand side of the interactive window allows to change the recipe parameters and re-execute the it. One can re-use values for for subsequent recipe executions by clicking the button ‘Use the parameters as initial values in subsequent executions of this recipe”, and then press “Continue WKF”. Note that the changes will be lost once reflex is closed. To save them in reflex, please save the workflow by clicking on “File” and “Export as xml” in the main reflex canvas.

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.
7.2 Computation of atmospheric model: running `molecfit_model` on selected spectra

The actor Model the atmosphere (`molecfit_mode`) is dedicated to compute the column densities of the various molecules in the atmosphere. This task is achieved by running the recipe `molecfit_model` onto a reference spectrum.

If the dataset contains only the science spectrum to correct, the `molecfit_model` recipe is executed on it. If the dataset contains a telluric standard star associated to the science spectrum we want to correct, then the user can fit either the telluric star or the science, depending on how the workflow parameter "Use science spectrum for fitting the atmosphere" has been set.

Before the fit, the spectrum is normalized to its median value. The normalization is done in the actor “Select and normalize science or STD MODEL for atmospheric model”.

An interactive window is associated to this execution step and it pops up once the recipe has been successfully executed (Figure 7.1). The actor inspect the input data and, if the ParameterInitialization is set to true, then it automatically sets some recipe parameter according to the selected instrument.

Moreover, independently from the ParameterInitialization, the actor understands the format of the spectrum to fit and sets the following recipe parameters accordingly:

- `USE_ONLY_INPUT_PRIMARY_DATA`.
- `USE_DATA_EXTENSION_AS_DFLUX`.
- `USE_DATA_EXTENSION_AS_MASK`.
- `COLUMN_LAMBDA`.
- `COLUMN_FLUX`.
- `COLUMN_MASK`.

These recipe parameters can, however, be set differently by the user.

The plot area shows the input extracted 1D spectrum in blue, and the best fit model obtained by molecfit in red. The wavelength regions used by the fit are highlighted in green. They are editable via direct input on the recipe parameter area (see also 7.2.1). Tool tip information displaying the best fit parameters and atmospheric model are prompted by moving the mouse cursor on the top of the plots.

The right-hand side of the interactive window allows to change the recipe parameters and re-execute the it. You can re-use your preferred parameter set as initial values for subsequent recipe executions by clicking the corresponding button. Once the optimal set of parameters is identified, press “Continue Wkf” to continue with the reduction. After having executed the recipe with the optimal set of parameters, you can store them and re-use for other executions: select “Use the parameters as initial values in subsequent executions of this recipe”, and then press “Continue WKF”. Note that the changes will be lost once reflex is closed. To save them in reflex, please save the workflow by clicking on “File” and “Export as xml” in the main reflex canvas.

Note that, if ParameterInitialization is set to true, then some recipe parameters are re-initialized independently of the set-up saved by the user (see Section 5.2 for more information).
7.2.1 Tips for an efficient fit.

The default parameters should provide a good model of the input spectrum in the majority of the cases. Here we list some tips to improve the fit or the recipe efficiency if the user needs it.

- **Wavelength ranges.** The fit is performed on a sub-set of wavelength ranges and not to the entire spectrum; this is found to be more efficient and less time-consuming. The advice is to select few wavelength ranges (at most 4 or 5 ranges) that include the expected molecules that are observable in the spectrum (see Figure 7.2 for an atlas). It is also advisable to define narrow wavelength regions, so that the continuum can be approximated by a 1st or 2nd order polynomial. If possible, it is recommended to include a range at the beginning and at the end of the spectrum, so that the entire spectrum wavelength range is bracketed. Telluric features that are too deep (e.g., close to saturation) should be avoided.

At the moment, the ranges to be fit can be inserted by direct input to the recipe parameter `WAVE_INCLUDE`. If this parameter is set to “NULL” (default), then the values will be read by a static configuration file (category: `WAVE_INCLUDE`).

Future releases will allow to specify the wavelength ranges by clicking with the mouse in the plot window. For example, the entry ‘0.815,0.830,0.972,0.986’ will perform the fit in the two ranges $0.815 < \lambda[\mu m] < 0.830$ and $0.972 < \lambda[\mu m] < 0.986$. Please, include the quotation marks “’”, do not include spaces.
- **Molecules.** In the fit, only the molecules that appear in the wavelength region of the spectrum should be fitted. The recipe parameters `REC_COL`, `FIT_MOLECT`, and `LIST_MOLEC` should then be defined accordingly. Note that an entry of 0 in `FIT_MOLEC` will not fit that specific molecule but will compute it. Its computation depends on the input `REL_COL`.

For a full list of supported molecules, please consult the Molecfit user manual [2].

- **Continuum and Wavelength Calibration.**

  The recipe parameters `FIT_CONTINUUM` (true/false) `FIT_WLC` (true/false) enable or disable the fit of the continuum and wavelength solution.

  The recipe parameters `CONTINUUM_N` and `WLC_N` set the order of the polynomial fit for the continuum and wavelength solution.

  The recipe parameters `CONTINUUM_CONST` and `WLC_CONST` set the initial values for the zero-th order coefficients. It is advisable to set `WLC_CONST`=0 (default). `CONTINUUM_CONST` is automatically set by the workflow to the median of the input spectrum to fit.

  It is advisable to fit both the continuum shape and the wavelength solution. Each wavelength region used in the fit has the continuum fitted independently; it is recommended to use a small polynomial degree such as 1 or 2 (unless the extension of the wavelength regions justifies the use of larger values). A global wavelength solution is used for all the wavelength regions. The wavelength solution is found by comparing the position of the telluric features in the observations with those of the internal molecfit atlas. Note that the output spectrum will not be adapted to the new wavelength scheme, in order to avoid extrapolation problems at the edges of the wavelength region. It is advisable to use a small polynomial degree (1 to 3) and insert wavelength regions at the edges of the spectrum (if telluric features are present).

- **Instrumental spectral resolution (kernel).**

  The fit to the observations accounts for the instrumental spectral resolution in several ways. The default is to use a simple Gaussian fit, with full with half maximum (in pixel) constant at all wavelengths. The relevant recipe parameters are:

  - `FIT_RES_BOX`: Fit resolution by Boxcar LSF. Default=FALSE.
  - `RES_BOX`: Initial value for FWHM of Boxcar rel. to slit width at the centre of the spectrum. If `FIT_RES_BOX`=FALSE this value will be fixed in the model (unless set to 0). Default=0.
  - `FIT_RES_GAUSS`: Fit resolution by Gaussian. Default=TRUE.
  - `RES_GAUSS`: Initial value for FWHM of the Gaussian in pixels at the centre of the spectrum. If `FIT_RES_GAUSS`=FALSE this value will be fixed in the model (unless set to 0). Default=0.
  - `FIT_RES_LORENTZ`: Fit resolution by Lorentzian. Default=FALSE.
  - `RES_LORENTZ`: Initial value for FWHM of the Lorentz in pixels at the centre of the spectrum. If `FIT_RES_LORENTZ`=FALSE this value will be fixed in the model (unless set to 0). Default=0.
  - `VARKERN`: If true, the resolving power is assumed constant (hence the instrumental FWHM varies accordingly).

  It is recommended to use small initial guesses for the instrumental FWHM, from 0.5 to 2 pixels.
Figure 7.2: Synthetic absorption spectrum of the sky between 0.3 and 2.6 microns calculated with LBLRTM (resolution $R= 10000$) using the annual mean profile for Cerro Paranal (Noll et al. 2012), adapted from Smette et al. 2015, A&A, 576, 77. The eight main molecules $O_2$, $O_3$, $H_2O$, $CO$, $CO_2$, $CH_4$, OCS, and $N_2O$ contribute more than 5% to the absorption in some wavelength regimes. The red regions mark the ranges where they mainly affect the transmission, minor contributions of these molecules are not shown.

7.3 Computation of atmospheric transmission: running molecfit_calctrans

Once the parameters of the atmosphere are computed (either from a telluric standard or from a science frame), the workflow combines them to determine the full telluric correction. The process accounts for the airmass difference between reference spectrum that was used to model the atmosphere (either telluric standard or science) and the science observations to be corrected.

This is done by the actor Construct telluric correction (molecfit calctrans). The actor reads the input file format and sets the following recipe parameter accordingly:

- USE_ONLY_INPUT_PRIMARY_DATA
- USE_DATA_EXTENSION_AS_DFLUX
- USE_DATA_EXTENSION_AS_MASK
- CALCTRANS_MAPPING_KERNEL
- MAPPING_ATMOSPHERIC
- MAPPING_CONVOLVE

This actor displays the interactive window shown in Figure 7.3.
7.4 Correction for atmospheric transmission: running `molecfit_correct`

Once the full telluric corrections are computed by the caltrans step, the workflow applies them to the science spectra. This step is done by the actor `correct data` that is visible in the main canvas and associated to the recipe `molecfit_correct`. Their interactive features are identical and they are described here and shown in Figure 7.4. The actor reads the input file format and sets the following recipe parameter accordingly:

- `USE_ONLY_INPUT_PRIMARY_DATA`
- `USE_DATA_EXTENSION_AS_DFLUX`
- `USE_DATA_EXTENSION_AS_MASK`
- `MAPPING_CORRECT`
Figure 7.4: The interactive window for the Telluric correction (recipe: molecfit_correct).
8 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.

- **After a successful reduction of a data set, I changed this data set in some way (e.g. modified or removed some files, or changed the rules of the Data Organizer). When I restart Reflex, the Data Set Chooser correctly displays my new data set, but marks it as “reduced ok”, even though it was never reduced before. What does this mean?**
  
  The labels in the column “Reduced” of the Data Set Chooser mark each dataset with “OK”, “Failed” or “-”. These labels indicate whether a data set has previously successfully been reduced at least once, all previous reductions failed, or a reduction has never been tried respectively. Data sets are identified by their name, which is derived from the first science file within the data set. As long as the data set name is preserved (i.e. the first science file in a data set has not changed), the Data Organizer will consider it to be the same data set. The Data Organizer recognizes any previous reductions of data sets it considers to be the same as the current one, and labels the current data set with “OK” if any of them was successful, even if the previously reduced data set differs from the current one.

  Note that the Product Explorer will list all the previous reductions of a particular data set only at the end of the reduction. This list might include successful and/or unsuccessful reduction runs with different parameters, or in your case with different input files. The important fact is that these are all reductions of data sets with the same first raw science file. By browsing through all reductions of a particular raw science file, the users can choose the one they want to use.

- **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 ([3]). 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>/</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 ([3]) 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 ([3]) 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 automatically 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.

• How to insert negative values into a textbox?
  Due to a bug in wxPython, the GUI might appear to freeze when attempting to enter a negative number in a parameter’s value textbox. This can be worked around by navigating away to a different control in the GUI with a mouse click, and then navigating back to the original textbox. Once focus is back on the original textbox the contents should be selected and it should be possible to replace it with a valid value, by typing it in and pressing the enter key.

• I’ve updated my Reflex installation and when I run esoreflex the process aborts. How can I fix this problem?
  As indicated in Section 2, in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the $HOME/KeplerData, $HOME/.kepler directories if present, to prevent possible aborts (i.e. a hard crash) of the esoreflex process.

• How can include my analysis scripts and algorithms into the workflow?
  EsoReflex is capable of executing any user-provided script, if properly interfaced. The most convenient way to do it is through the Python actor. Please consult the tutorial on how to insert Python scripts into a workflow available here: www.eso.org/sci/data-processing/Python_and_esoreflex.pdf
9 Troubleshooting

![The Data Organizer interactive window reports an error “:No DataSets have been created, check the data set and the OCA rules.”](image)

Figure 9.1: The Data Organizer 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 crashes**

   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 uncorrupted 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 9.1.)

   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 evaluate the quality.

At the bottom right 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 Forchi, Wolfram Freudling, César Enrique García, Maurice Klein Gebbinck, Andrea Modigliani, Sabine Moehler, Mark Neeser & Martino Romaniello.