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

Molecfit Reflex Tutorial

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

Issue 4.3

Date 2022-05-??

Prepared: J. Pritchard, L. Cocccato, N. Fernando, N. P. F. Lorente, A. Smette, M. Birc
Name

Approved: –
Name

Released: –
Name

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section affected	Reason/Initiation/Documents/Remarks
3.0	03/06/2020	All	A new Molecfit Pipeline has been released
4.0	01/08/2020	1, 3, 5-7	New workflow version; support for XSHOOTER and UVE
4.1	05/02/2021	5, 7	Details on improved example workflows, and chips & rang changes.
4.2	05/12/2021	5, 7	Details on new workflow, and new Demo data.
4.3	05/2022	1, 3, 5.2, 6, 7, 8.6.2, Appendices	Updates to reflect improved user interface `molecfit_model` GUI.

This page was intentionally left blank

Contents

1 Introduction to `Esoreflex` 9

2 Software Installation 11

2.1 Installing Reflex workflows via `macports` . . . 11

2.2 Installing Reflex workflows via `rpm/yum/dnf` . . . 11

2.3 Installing Reflex workflows via `install_esoreflex` . . . 12

2.4 Demo Data . . . 13

2.5 Before starting the Quick Start Demo . . . 13

2.5.1 Background execution . . . 13

2.5.2 Available Disk Space . . . 14

3 Quick Start: Reducing The Demo Data 15

4 About the main `esoreflex` canvas 19

4.1 Saving And Loading Workflows . . . 19

4.2 Buttons . . . 19

4.3 Workflow States . . . 19

5 The Molecfit Workflow 20

5.1 Workflow Canvas Parameters . . . 20

5.2 MOLECFIT-specific workflow parameters . . . 21

5.3 Workflow Actors . . . 22

5.3.1 Simple Actors . . . 22

5.3.2 Lazy Mode . . . 22

6 Reducing your own data 24

6.1 The `esoreflex` command . . . 24

6.2 Launching the workflow . . . 24

6.3 Workflow Steps . . . 26

6.3.1 Data Organisation And Selection . . . 26

6.3.2 DataSetChooser . . . 27

6.3.3 The workflow data-reduction cascade . . . 28
6.3.4 The ProductExplorer . . . 29
6.4 Input file formats . . . 31
6.4.1 Intrinsic molecfit input file formats . . . 31
6.4.2 Molecfit worflow supported input file formats . . . 32
6.5 Description of the final Molecfit data . . . 33
6.5.1 Temporary and End products for demo datasets . . . 33

7 Optimising Your Results Through Workflow Interaction 35
7.1 General characteristics of the interactive windows . . . 36
7.2 Computation of atmospheric model: running `molecfit_model` on selected spectra . . . 38
7.2.1 Tips for an improved fit. . . . 41
7.2.2 Experienced/Expert user level tips for an improved fit. . . . 49
7.2.3 Use the above parameters as initial values... . . . 52
7.2.4 Apply ... Parameters buttons . . . 54
7.3 Computation of atmospheric transmission: running `molecfit_calctrans` . . . 54
7.4 Correction for atmospheric transmission: running `molecfit_correct` . . . 56

8 Correcting other instrument data 58
8.1 Correcting ESPRESSO data . . . 58
8.1.1 Supported input file formats . . . 58
8.1.2 Optimizing . . . 58
8.2 Correcting GIRAFFE data . . . 60
8.2.1 Supported input file formats . . . 60
8.2.2 Optimizing . . . 60
8.3 Correcting UVES data . . . 64
8.3.1 Supported input file formats . . . 64
8.3.2 Optimizing . . . 64
8.4 Correcting VISIR data . . . 67
8.4.1 Supported input file formats . . . 67
8.4.2 Optimizing . . . 67

8.5 Correcting XSHOOTER data . . . 68
8.5.1 Supported input file formats . . . 68
8.5.2 Optimizing . . . 68
8.6 Correcting ANY set of data . . . 72
8.6.1 Supported input file formats . . . 72
8.6.2 Optimizing . . . 72
8.6.3 Recipe parameters . . . 73
8.7 Adding support for other a.k.a. Custom instruments . . . 75
8.7.1 Custom instrument OCA rule files . . . 75
8.7.2 Custom instrument python modules . . . 76

9 Frequently Asked Questions **77**

10 Troubleshooting **80**

A Auto-Molecule **82**

1 Introduction to `Esoreflex`

This document is a tutorial designed to enable the user to to reduce his/her data with the ESO pipeline run under an user-friendly environmet, 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 have a look to https://support.eso.org to see if this has been reported before or open a ticket 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, using the Molecfit `esoreflex` molecfit.xml workflow.

Molecfit models the atmosphere by fitting regions of an observed spectrum and then generates the transmission correction for the complete wavelength range of a given science spectrum. 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.

Unlike the standard instrument workflows which allow the processing of RAW science data into pipeline products, as described above, the Molecfit `esoreflex` molecfit.xml workflow processes pipeline product files to correct for effects of the atmosphere. In general the workflow can process products **directly generated by running the pipeline(s)**[2], hereafter *Instrument Pipeline Products* [IPPs] **as well as Advanced Data Products** [ADPs] obtained from the ESO Science Archive Facility.

Another 'feature' of the Molecfit `esoreflex` molecfit.xml workflow that distinguishes it from standard instrument workflows is that it (aims to) allow the processing of data from multiple instruments (in principle *any* instrument, including non-ESO instruments). To enable this it relies on a number of python scripts to reformat input files – if necessary – into temporary working versions that are compatible with the molecfit recipes. Further python scripts are then executed after running the molecfit recipes to reformat the molecfit output back into the format of the original input files.

[1] *http://www.eso.org/sci/archive/calselectorInfo.html*

[2] But see section 6.4 for details and caveats.

In principle, the current version allows the telluric correction of the majority of ESO instruments, as well as non-ESO instruments, provided that the user sets the recipe parameters according to the instrument set-up, file format, and wavelength range, but this is not always trivial for non-expert users.

The workflow thus also offers 'full' support for a number of instruments, currently ESPRESSO (1D spectral format only), GIRAFFE, UVES (RED arm), VISIR, and XSHOOTER (all instrument modes except `IFU` and the `UVB` arm). For these *supported instruments* the workflow will automatically configure the workflow parameters so that it can be run *out of the box* for data reduced by the instrument pipelines or downloaded from the ESO science archive. We plan to add full support for more ESO instruments in due course.

Expert users can adapt the workflow to support other instruments. Section 8.6 describes the `INSTRUMENT=ANY` mode (useful if you have just a few spectra to correct), while section 8.7 describes how to implement support for custom/user instruments.

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

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 10.14 to 11, while the `rpm/yum` repositories support Fedora 28 to 32, 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 2.11.x 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 28 to 32, 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, run the following commands:

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

   - 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
     sudo yum-config-manager --add-repo=ftp://ftp.eso.org/pub/dfs/
               pipelines/repositories/stable/centos/esorepo.repo
     ```

- 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
sudo yum-config-manager --add-repo=ftp://ftp.eso.org/pub/dfs/
    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)
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)
sudo yum install esopipe-xshoo-all # (CentOS 7, SL 7)
```

- To install all pipelines use:

```
sudo dnf install esopipe-\*-all # (Fedora)
sudo yum install esopipe-\*-all # (CentOS 7, SL 7)
```

For further information, please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/rpm.html.

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:

```
wget https://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. 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 recieve demo data sets for each fully supported instrument (see section 1) that allows 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.

2.5 Before starting the Quick Start Demo

2.5.1 Background execution

The `molecfit_model` recipe can take a long time (tens of minutes, hours even...) to execute, depending on your hardware. It can therefore be more efficient to 'pre-run' the workflow non-interactively (whilst you go and have a coffee) on the data before sitting down to the workflow interactively. This will compute the first solutions, meaning that when you run the workflow the second time, interactively, thanks to the `Esoreflex` *lazy-mode* it doesn't actually have to re-compute anything for the first solution. Only if you change the recipe parameters as suggested in section 7 will the workflow then need to actually run the recipes again, and of course that will then take the time it takes...

Because of some of the features specific to the molecfit workflow, several extra command line options need to be specified to run the workflow non-interactively, compared to other standard pipeline workflows. The following command will pre-run the workflow in preparation for starting the next section, Quick Start.

```
esoreflex molecfit -n -INSTRUMENT XSHOOTER \
   -CheckInitializationMode No \
   -UserParamInit true \
   -SelectDatasetMethod All
```

Of course XSHOOTER can be replaced with any of the other supported instruments (ESPRESSO, GIRAFFE, UVES, VISIR) to pre-run the workflow for one of those.

2.5.2 Available Disk Space

The amount of diskspace *temporarily* needed by the `molecfit_model` recipe depends principally on the number of wavelength data points within the fitting range(s). In some cases this can amount to more than 10Gb. Please insure you have sufficient free diskspace available on the disk where the `ROOT_DATA_DIR` points to (by default this is `$HOME/reflex_data`), we recommend you have a minimum of 12Gb of free disk space. The commandline option *-ROOT_DATA_DIR* can be used to relocate `ROOT_DATA_DIR`, e.g.:

```
esoreflex molecfit -n -INSTRUMENT XSHOOTER \
   -CheckInitializationMode No \
   -UserParamInit true \
   -SelectDatasetMethod All \
   -ROOT_DATA_DIR /my_big_disk/reflex_data
```

or this can be done interactively from the main canvas.

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[3]. 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.

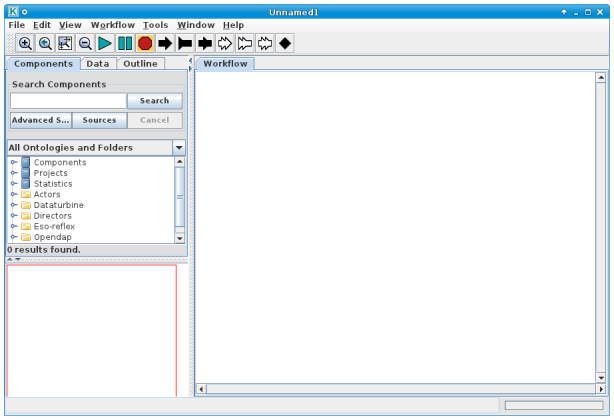


Figure 3.1: The empty `Reflex` canvas

Note 1: It is possible to configure the data modelling strategy by setting appropriate parameters in the main canvas. This will be explained in Section 5.2

[3]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.

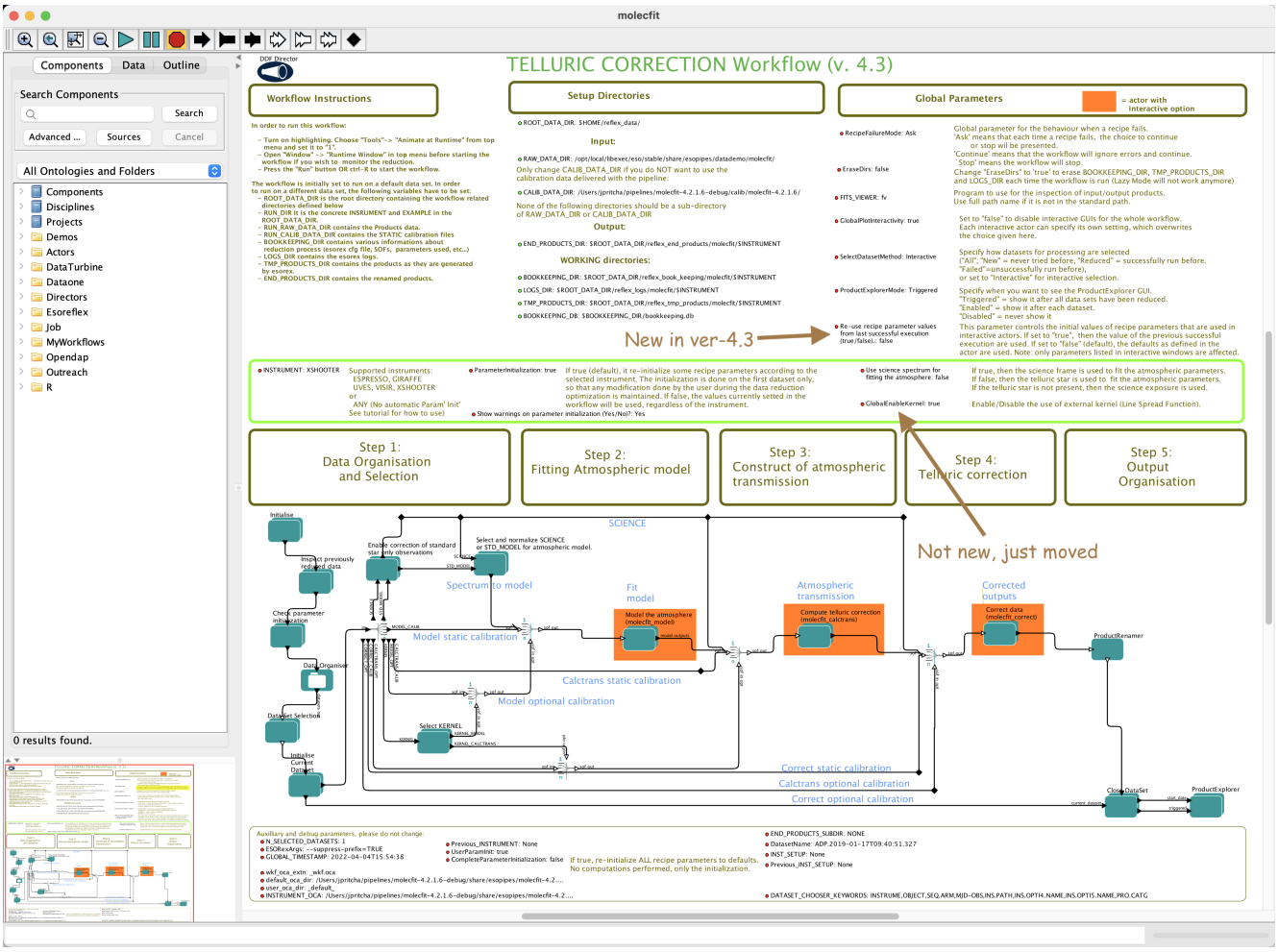


Figure 3.2: Molecfit workflow general layout.

Note 2: When the workflow starts, the user will be warned that some recipe parameters will be (re-)initialized according to the choosen instrument and the instrument setup used for the observations. For the demo dataset, the user can continue with the processing by simply accepting[4] 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. The first interactive window is actually displayed **before** executing the first recipe. This gives you the opportunity to make a first attempt to optimise some the fitting parameters. A little effort at this first instance can make a huge difference in the time it takes to optmise the model. The same GUI will then re-appear (at least) one more time after the recipe has executed giving you the chance to further optimise parameters in order to obtain a better model fit and re-execute the recipe, or to continue on to the next steps.

[4]Click the 'Yes' button. Clicking 'No' will stop the workflow.

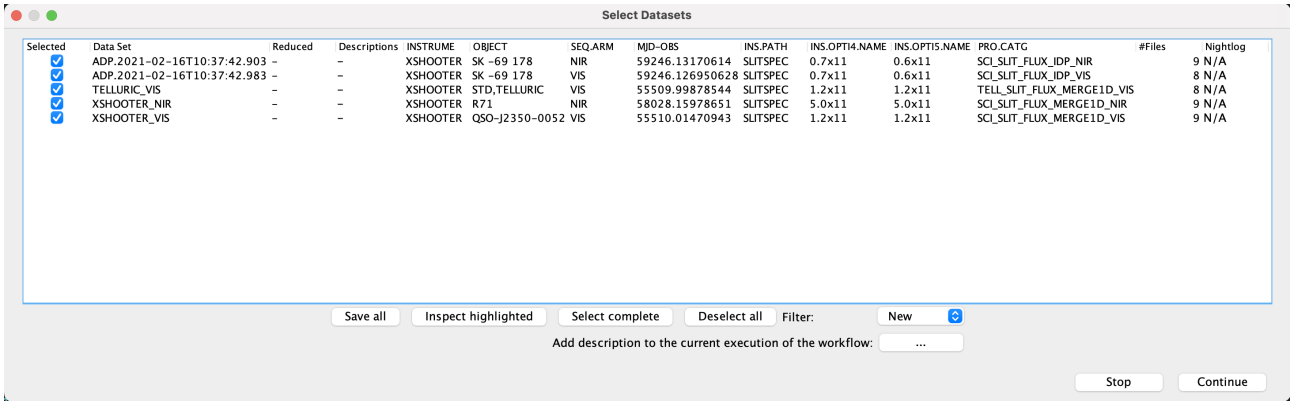


Figure 3.3: The 'Select Datasets' pop-up window for the XSHOOTER datademo package data, after being resized to display all information.

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; [4]).

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 Molecfit workflow main canvas contains several parameters that are specific to Molecfit. They are located in the green box in the middle of the canvas spanning the full width of the canvas. They allow the user to select the configure the general parameters of the workflow and control aspects of data processing 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 consider for processing. Available options are: ANY, ESPRESSO, GIRAFFE, UVES, VISIR, and XSHOOTER. Each selection selects from the input directory only the data from the instrument it refers to. The workflow can automatically sets some recipe parameters accordingly to the selected instrument (see **ParameterInitialization**). The option ANY does not make this distinction and selects data also for the instruments for which there is no parameter initialization support, as long as the data fulfill some requirements (see Section 8.6). When selecting ANY, it is advisable to turn off the automatic parameter initialization (see below) so that each parameter can be configured manually in the appropriate interactive windows (see Section 7).

- **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.

- **Show warnings on parameter initialization**. If set to `Yes`, a popup window is displayed when the workflow is run, essentially before doing anything, and in particular before making any changes to any of the parameters stored in the workflow itself, warning the user that changes will (or will not – according to the value of **ParameterInitialization**) be made by the workflow scripts, according to the instrument and instrument setting. The user is asked if they wish to proceed and must click the "Yes" button to do so. If set to `No`, no popup window is presented and the workflow continues.

- **Use the science spectrum for fitting the atmosphere**. If `true`, then the science frame is used to fit the atmospheric parameters. If `false`, then *if* a telluric star frame is associated to the science frame as part of the data-set then the telluric star frame 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 show 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 science spectrum to correct (e.g., must be observed with the same instrument configuration and on the same night as the science spectrum to correct).

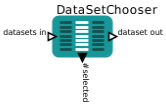
- **GlobalEnableKernel** : `true`/`false`: Enable/Disable the use of calibration files describing the instrument line spread function (kernel). The current molecfit workflow release does not include any kernels. If no kernel is provided, this option has no effect on the processing. Default: true.

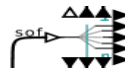
5.3 Workflow Actors

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:

-

- The `DataOrganiser` actor.
- 

- The `DataSetChooser` actor (inside a composite actor).
- 

- The `FitsRouter` actor Redirects files according to their categories.
-

- The `ProductRenamer` actor.
-

- The `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/reflex_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[5] 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 `RecipeExecuter`. The file categories are used by the `FitsRouter`

[5]OCA 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[6], 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.

[6] keep the mouse pointer on the file name to visualize the full path name.

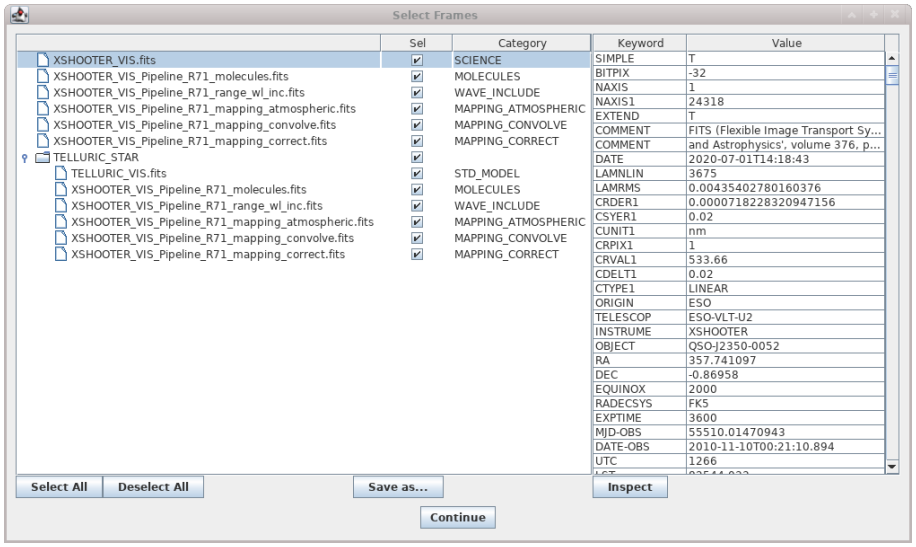
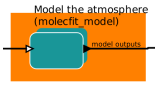


Figure 6.1: The 'Selected Frame' pop-up window, obtained after pressing Inspect highlighted in the Select Datasets window (Fig. 3).

6.3.3 The workflow data-reduction cascade

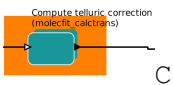
The present Molecfit workflow is designed to process the datasets according to a specific data reduction cascade. This cascade triggers a series of pipeline recipes, which are associated to the following composite actors:

- 

`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 [3] 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).

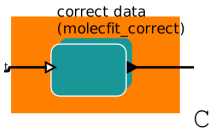
  This is an interactive actor, meaning that the results are displayed in an interactive window for inspection. The user has the possibility to modify the recipe parameters and re-run this step. The interactive features are described in Section 7.2.

- 

`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 the complete 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 [3] for further information.

This is an interactive actor, meaning that the results are displayed in an interactive window for inspection. The the user has the possibility modify the recipe parameters and re-run this step, though actually there is very little the user can alter at this stage, all the optimisation is done at the `molecfit_model` stage. 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 in an interactive window for inspection. The the user has the possibility modify the recipe parameters and re-run this step, but again there is very little the user can alter at this stage, all the optimisation is done at the `molecfit_model` stage. The interactive features are described in Section 7.4.

We refer the user to the Molecfit User manual [3] for a complete description of the recipes and their parameters.

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 6.5. The intermediate pipeline calibration products can be found in subdirectories of the `TMP_PRODUCT_DIR`.

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 datareduction:

- 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 attemps. Green and red flags identify successfull or unsucessfull 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 succesful (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:

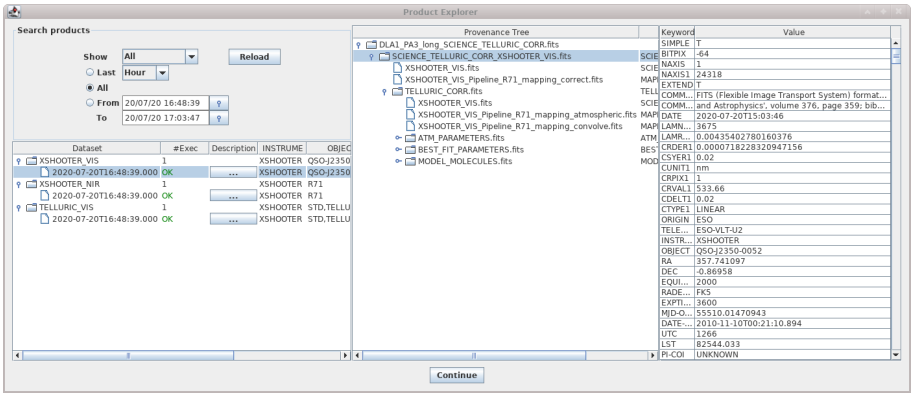


Figure 6.2: The 'Product Explorer' pop-up window showing final results of the workflow.

* Copy full path. It copies the full name of the file onto the clipboard. Shift+Ctr+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 Input file formats

6.4.1 Intrinsic molecfit input file formats

Intrinsically the molecfit recipes support two types of FITS files[7] as input data, or more precisely two types of FITS file Header Data Units [HDUs] (see below). It is however not possible to have input files consisting of both types of HDUs, thus in practice it really is two types of FITS files.

The molecfit recipes have as a minimum requirement input wavelength and flux data. Error value and mask value data (at each wavelength point) can optionally be provided. All data (required and optional) must be found in a single input file.

The input FITS file thus must contain either:

[7] ASCII input files are no longer supported since version 3 of molecfit.

- **One dimensional Image HDUs**: with the image pixel values representing the flux values and the wavelength calibration specified via standard WCS header information (e.g. CRPIX, CRVAL, CDELT). Flux, and optional error and/or mask values can/must be specified via multiple extensions. The extension(s) to be used for each value type are fully configurable via the input recipe configuration files.
- **Binary Table HDUs**: with columns providing wavelength, flux and optionally errors and/or mask values. The names of the cloumns to be used are fully configurable via the input recipe configuration files.

See the Molecfit User manual [3] for more details.

6.4.2 Molecfit worflow supported input file formats

For the supported instruments the workflow has been implemented to support the **standard output format(s) of the respective ESO instrument pipelines**, hereafter *Instrument Pipeline Products* [IPPs] *and* **Advanced Data Products** [ADPs] obtained from the ESO Science Archive Facility[8].

The format of ADPs is such that they are supported intrinsically by molecfit. Some IPPs are also supported intrinsically (e.g. UVES and ESPRESSO) while for others instrument specific python scripts[9] reformat the non molecfit intrinsically supported IPPs into one of the molecfit intrinsically supported formats for processing by molecfit. Other instrument specific python scripts reformat the molecfit products back into the original format of the input data. This is all handled internally by the workflow and is thus transparent to the user. See the instrument specific sections in section 8 for details about how reformatimg is done (when necessary.)

You can thus point the molecfit workflow[10] to a reflex_end_products directory produced by running the respective instrument pipeline on instrument RAW data. However because esoreflex was originally designed to run on instrument RAW data it does not expect (and can not handle) multiple input files with the same filename. Unfortunately many of the instrument pipelines will generate multiple end products with identical names, for example when:

- the instrument pipeline workflow is run more than once on the same dataset.
- a single OB (or multiple OBs with the same OB name) executes multiple exposures, generating multiple RAW FITS files which are treated as distinct datasets and thus processed individually by the instrument workflow and given the *de facto* standard filenames based on OB name. UVES and the UVES workflow behave this way. On the otherhand, while GIRAFFE also behaves this way the GIARFFE workflow includes the datetime stamp of the start of the RAW file[11] in the end product filenames thus making each filename unique (except when the pipeline is run multiple times...).

One thus needs to take some care in choosing which files are 'exposed' to the workflow. Remember that esoreflex does a recursive search for FITS files in the directory specified by the RAW_DATA_DIR variable on the main canvas.

[8] Some pipeliens also produce ADP format files directly, in which case the IPPs are infact ADPs too!

[9] With this concept is it thus, in principal, possible to 'support' any input file format, provide the appropriate 'python file reformatter script/algorithm' can be provided.

[10] i.e. set the RAW_DATA_DIR variable on the main canvas

[11] When multiple files are grouped together for processing by the GIRAFFE workflow the datetime stamp of the start of the *first* RAW file is used

On the otherhand, it is no problem to mix up data of various file formats and even instruments in the same RAW_DATA_DIR directory (or sub-directories thereof).

6.5 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.

6.5.1 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 product category is `SCIENCE_TELLURIC_CORR`, as defined by the header keyword: `HIERARCH ESO PRO CATG`. The formats of the product files 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 wavelenght 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_calctrans.
  * 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 dataset supplied with the installation to illustrate how to optimise the scientific products in terms of quality and signal-to-noise ratio.

There are also a number of videos available on the web aimed at providing guidance on how to optimise molecfit. ESO maintains a Youtube Playlist of Molecfit related video tutorials[12].

As a general rule however, it should be stressed that the automatically set parameters should always be considered as a *hopefully* useful first 'guess'. In probably most, if not all real world use cases it will almost certainly be necessary to adjust and fine tune at least some of the molecfit_model parameters, in particular the wavelength regions to fit (WAVE_INCLUDE) and exclude (WAVE_EXCLUDE) and to carefully consider the list of molecules (LIST_MOLEC) to include and fit.

The examples shown in this Section refer to the XSHOOTER_NIR dataset (object name: R71); information related to other datasets will be added whenever relevant.

This is work in progress and the contents of this section will be expanded 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 processing for all XSHOOTER 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. Selected datasets will be then processed in series.

To follow the rest of this section, select only the XSHOOTER_NIR dataset in the Dataset Chooser window, see Figure 7.1.

After clicking the **'Continue'** button in the Select Datasets window the selected datasets will be processed, with the files of each selected dataset being sent to the correct pipeline recipes, so that the data reduction can 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 interactive windows, allowing the user to inspect the products and, if desired, 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 with an associated interactive window crashes[13] and you decide to continue the workflow, an

[12] https://www.youtube.com/playlist?list=PLDNJqjce4cUxF2CM-FWuBTrUDNdugkLia

[13] Unfortunately, sometimes the recipes crash in such a way that the workflow simply stops with no option to continue, often with an

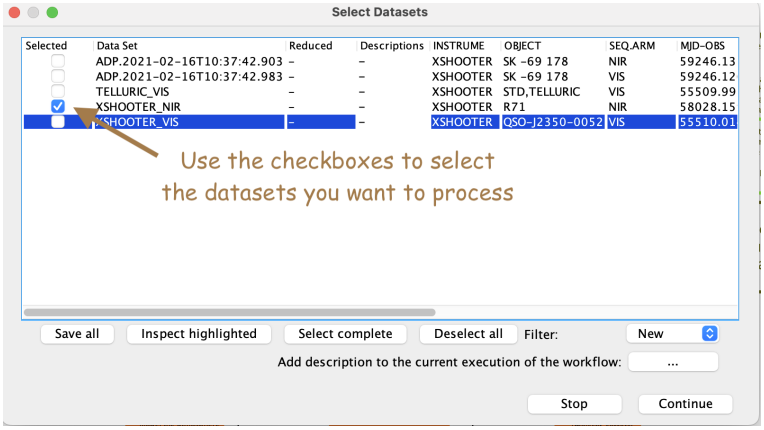


Figure 7.1: Select just the XSHOOTER_NIR dataset for processing.

empty interactive window will be displayed allowing to change the recipe parameters and re-execute it.

7.1 General characteristics of the interactive windows

For each of the interactive windows described above (section 6.3.3), 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.

unintelligible error message. In such cases, if you are not able to invesitgate the cause of the error yourself, please submit a ticket at https://support.eso.org.

- - 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.

On the the right-hand side of the interactive window is a panel with various components. At the top a ’notebook’ with one or more tabs presenting the recipe parameters that can be adjusted[14]. Below that, buttons to continue on to the next stage of the workflow or re-execute the recipe, and then Checkboxes that permit saving the current recipe parameters for future use and turning off display of that particular GUI for future datasets. And finally a listing of 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.

[14] In general, not all recipe parameters are presented, usually only the ones most commonly adjusted by the majority of users, if any adjustment is necessary. To adjust parameters not presented in the GUI, one must edit the values in the relevant actor before starting the workflow. See [4].

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

The `Fit model` actor is dedicated to computing the column densities of the various molecules in the atmosphere. This task is achieved by running the recipe `molecfit_model` on a spectrum.

This is essentially where all the 'work' of this workflow is done. One must 'define' the model at this stage. There is no possibility to change the model at the subsequent workflow steps (Calctrans and Correct) which simply apply the results calculated here.

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 main canvas workflow parameter **"Use science spectrum for fitting the atmosphere"** is set.

Before enterring the `Fit model` actor, the input data is inspected and converted into a molecfit compatible format if necessary[15] and then the spectrum is normalized to its median value[16].

Then, within the `Fit model` actor, if the ParameterInitialization is set to `true`, various recipe parameters are automatically set according to the instrument and instrument-setting of the current dataset.

Moreover, independent 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.

Recipe parameters can be set differently by the user. If ParameterInitialization is not `true` the following recipe parameters need to be set for each configuration accordingly:

- FIT_CONTINUUM.
- CONTINUUM_N.
- FIT_WLC.
- MAP_REGIONS_TO_CHIPS.

If the input SOF does not include a files with categories `MOLECULES` and/or `WAVE_INCLUDE`, auto-molecule (see appendix A) will be used to set initial values for the corresponding recipe parameters:

[15] In the composite actor labelled 'Enable correction of standard star only observations'.
[16] In the composite actor labelled 'Select and normalize SCIENCE or STD_MODEL for atmospheric model'

- WAVE_INCLUDE
- WAVE_EXLCUDE
- PIXEL_EXLCUDE
- LIST_MOLEC
- FIT_MOLEC
- REL_MOL

The above parameters will also be set according to the values in the files of the following tags if present in the input dataset:

- MOLECULES
- WAVE_INCLUDE
- WAVE_EXCLUDE
- PIXEL_EXCLUDE

Values from files will override auto-molecule values.

Setting these parameters to the values from the calibrations files, rather than setting the recipe parameters to `NULL` and allowing the `molecfit_model` recipe to read them directly from the files makes it easier to adjust the settings interactively if necessary.

New in ver. 4.3: **Before** launching the `molecfit_model` recipe, the `molecfit_model` GUI window is displayed a first time (Figure 7.2), allowing the user to already make a first attempt to optimise parameters before running the recipe for the first time.

The plot area shows the input spectrum used for fitting the model in blue. The wavelength regions used by the fit are highlighted in green. They are editable via direct input on the Recipe Parameter area (on the 'Expert' tab, see also 7.2.1) or via point-and-click interaction with the plot using the Add/Delete/Modify buttons under the 'Include regions' and 'Exclude regions' headings in the panel on the right.

New in ver. 4.3: The `molecfit_model` recipe parameters have been organised into four tabs:

- **Fit statistics**: (Only displayed after execution of `molecfit_model` recipe), provides information only about the fit
- **Molecules**: Most users will need to interact with these parameters
- **Experienced**: A few parameters that Experienced users of molecfit *might* want to adjust directly
- **Expert**: All other parameters that probably only Expert users of molecfit should think very carefully about adjusting directly

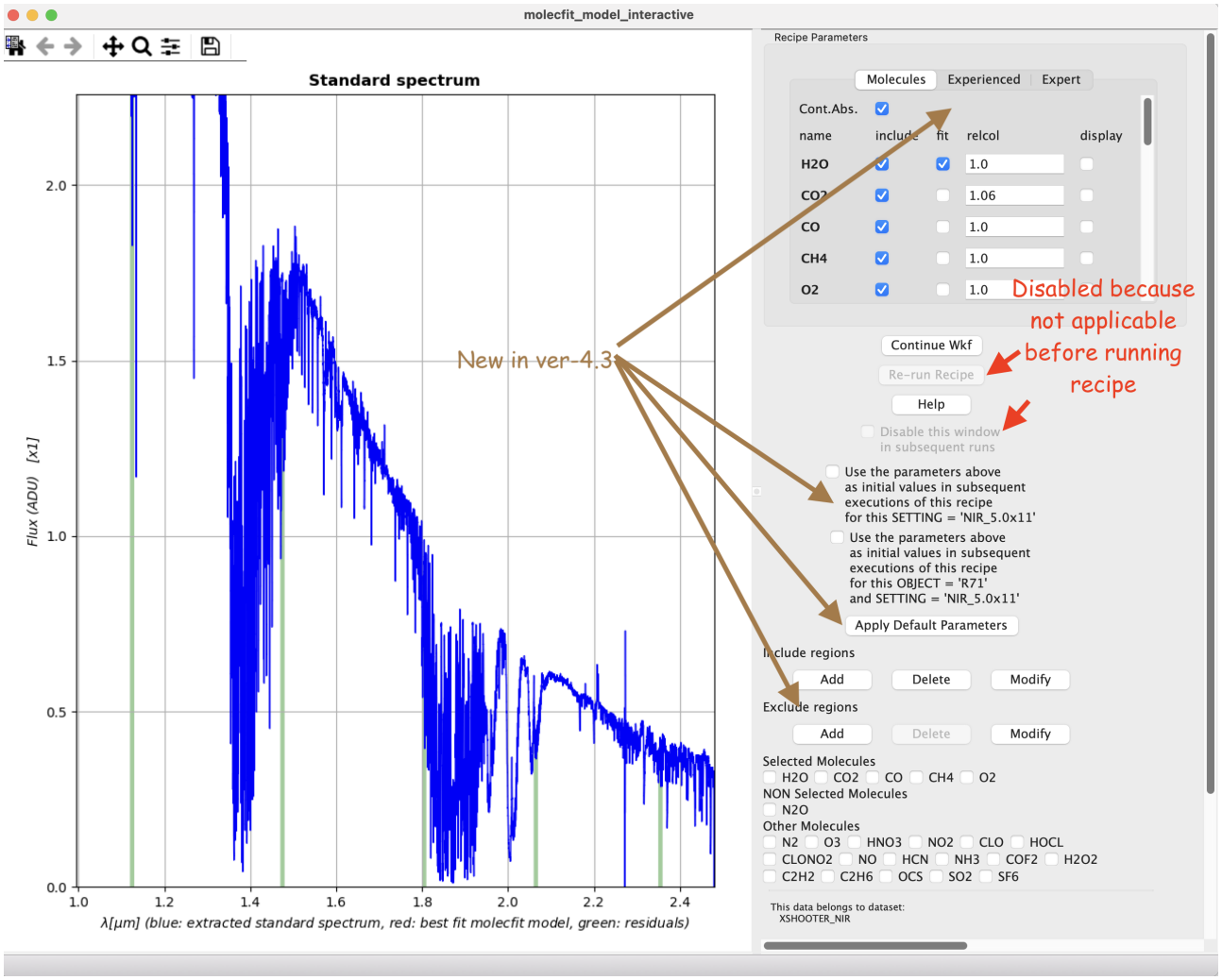


Figure 7.2: The interactive window displayed **before** attempting the atmospheric fit for the first time (recipe: *molecfit_model*).

The `molecfit_model` recipe can take a *long* time to execute[17], so it is defintiely worthwhile to spend some time to optimise the parameters even before the first execution. In particular, the automatic range setting algorithm (see Appendix A) often results in rather broad fitting ranges which for most science objects will inlcude spectral features from the object, rather than the earth's atmosphere. Object-spectral-features should not be incldued in the fitting – either by restricting the fit regions, or defining exclude regions within the fit regions.

New in ver. 4.3: What are the most useful parameters I can change before the execution of the recipe? It is advisable to select the molecule to consider in the model. One can either "fit" a given molecule, or "include" it (in this case, it's contribution will be computed according to the relations with other molecules). The molecules can be specified either in the Molecules tab, or in the Experienced tab (see Figure 7.2). Then, one should carefully select the wavelength regions to fit. This can be done either by specifying the wavelength boundaries in the Experienced tab, or by adding/deleting buttons in the interactive window. For example, to add a region: press Add, select the region by dragging the mouse in the plot (a green area will be highlighted), and then press Add again (see Figure 7.7). Same procedure for Deleting a region. One can identify the wavelength regions that are affected by a given molecule just by selecting the molecule in the bottom list of the interactive window. The contribution of that molecule will be shown in the plot (see Figure 7.6) so that one can judge what wavelength region and molecule to consider.. More detailed instructions and tips are given in Section 7.2.1.

In the case of XSHOOTER however, auto-molecule is not invoked, because moleules to include and fit and fitting ranges for VIS and NIR data have been defined and are set in each dataset via the appropriate category=MOLECULES and category=WAVE_INCLUDE files included in the molecfit static calibration package.

So let's run the model fitting with the parameters as they are, click the **'Continue Workflow'** button, and the recipe will now be executed, inless you already pre-processed the demodata (see section 2.5) in which case, thanks to Lazy mode, esoreflex will reuse the results from before and the next GUI will appear quickly. If not executing the `molecfit_model` recipe may take anywhere from a few minutes, to a few tens of minutes, depending on your hardware.

Once the recipe does complete, the interactive `molecfit_model` GUI window is displayed a second time now (Figure 7.3).

The plot area now shows the input spectrum used for fitting the model in blue, the best fit model within the fitted wavelength ranges obtained by molecfit in red and the residuals between data and model in green. The wavelength regions used by the fit are highlighted in green (and yellow).

So we have a fit, but how good is it?

New in ver. 4.3: Considering the information in the the Fit statistics tab of the Recipe Parameters notebook, there's no way to say on an absolute scale a χ^2 (or reduced-χ^2) of less than some value X is 'good' and greater than this is 'bad'. These statistics will depend on, amongst other things, the input dataset, the parameters being fit (in particular the number of molecules being fit, the number and size of fit wavelength regions). One simply has to put the time and effort to convince oneself – in a scientific manner – that better can't be done. The next section describes some ways to do that...

7.2.1 Tips for an improved fit.

[17] Depending on the number of points in the input spectrum, the number and spectral range of regions being fit, the number of parameters and molecules being fit, and of course the hardware it is running on, it can take 10's of minutes, 1 or more hours even...

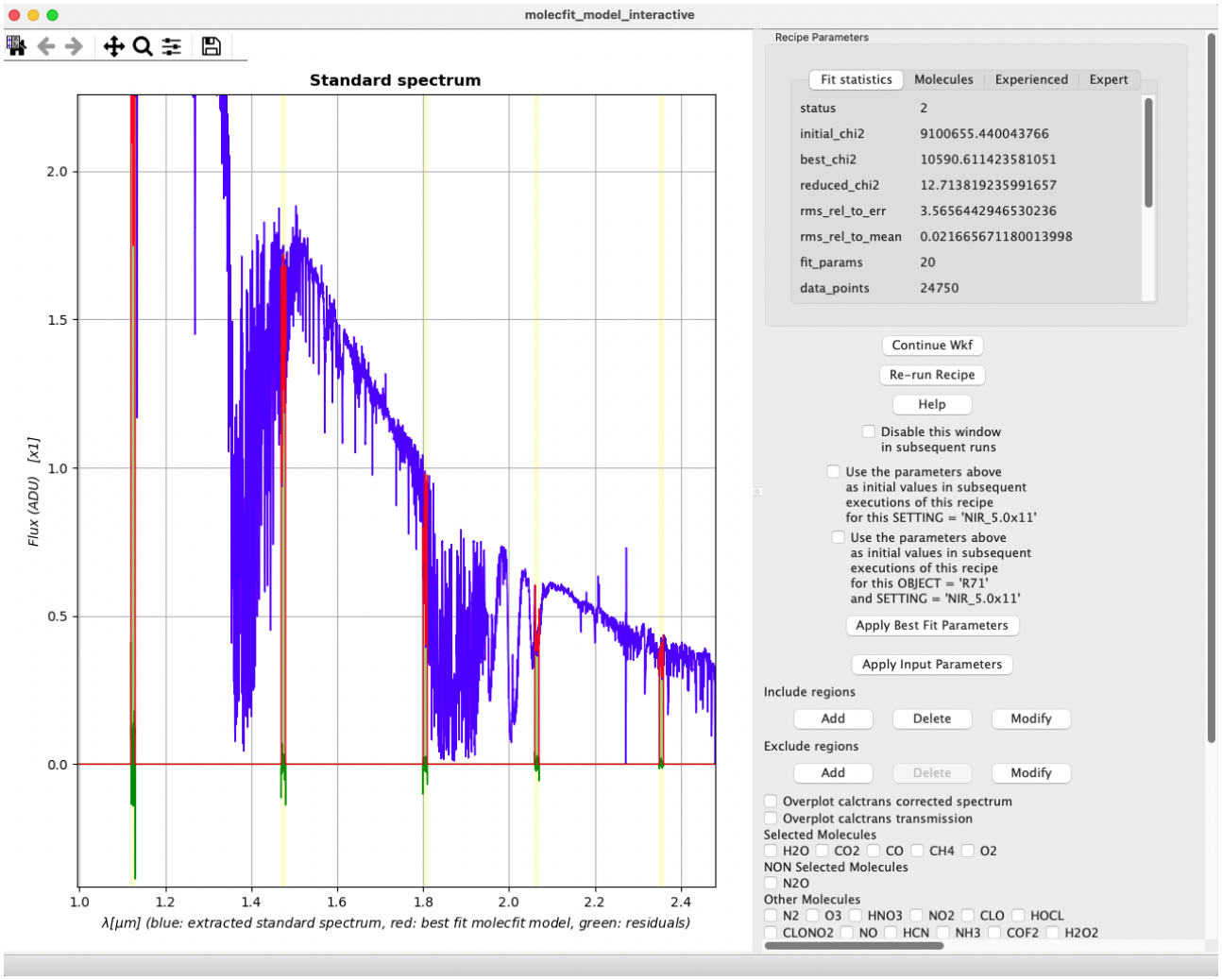


Figure 7.3: The interactive window displayed **after** the atmospheric fit (recipe: *molecfit_model*).

New in ver. 4.3: The default XSHOOTER settings are not always optimal. Indeed for the NIR XSHOOTER datasets, you can see in the 'Molecules' tab of the Recipe Parameter notebook (top right corner of the `molecfit_model` C while the five relevant molcules have been included in the model, only H2O has been fit, the others have been included with te REL_COL factors specified.

Before going further, it would be useful to see the entire spectrum. By default, the y-axis scale is set such that the spectrum below about $1.3\mu m$ is lost. You can interactively re-scale the plot display as follows:

- Click the Zoom icon (usually a magnifying glass) in the action bar top left of the window
- Holding the **right**-mouse button, and starting in the bottom left corner of the plot (i.e. coordinates around $[1.0, -0.4]$) drag a box to coordinates $\sim [2.46, 0.8]$
- Click the Zoom icon again to turn off zooming mode

After the above steps the display should look something like Figure 7.4.

Now we can include all five molecules in the fit simply by checking the checkboxes for the other four molecules, see Figure 7.5.

Lets try that and see if it improves the fit – keep in mind the 'best_chi2' and 'reduced_chi2' statistics from this first fit ($\sim$10590.5 & $\sim$12.7 respectively). Click the **Re-run Recipe** button to re-run the `molecfit_model` recipe with the new parameter settings.

The 'best_chi2' and 'reduced_chi2' statistics for the new fit including all five molecules should now be $\sim$10482.3 & $\sim$12.6 respectively. An improvement for sure, but is it significant and perhaps more importantly meaningful?

Switch to the Molecules tab, and then click the 'Apply Best Fit Parameters' button. This method behind this button reads the results of the `molecfit_model` recipe execution and applies them to the relevant parameters. The fitted values of 'REL_COL' (in the 'relcol' column) all look reasonable (near 1.0) except the one for O2 which is 88.2! That can't be right. What's going on?

Click the checkbox in the display column in the O2 row. This will overplot the atlas spectrum for O2. As you can see (in Figure 7.6), there are two wavelength regions with significant absorption from O2 but no fit regions covering any part of either off these two wavelength regions.

Obviously in order to make a meaningful fit to O2, we need to include at least one fitting wavelength region that includes absorption lines from O2.

Ideally fitting regions should be relatively 'short' but contain a 'significant' number of atmospheric lines, perferably without lines from one molecular species contaminating another. For O2, the range bewteen 1.23 and 1.29 μm is pretty good, with possibly just a little contamination from H2O. We could add this range to WAVE_INCLUDE in the Expert tab, but it is much easier to use the point and click method:

- Make sure zoom mode is off – if you hover the mouse over the action bar in the top left corner, the currently active 'mode', if any (pan/zoom or zoom rect) will be written at the right-hand end of the bar. If one of those modes is active, just click the corresponding button again to toggle it off.
- Click the 'Add' button in the row immediately under the 'Include regions' headerin the grey panel on the right of the window. Notice the plot title will change to 'LMB = Select region : RMB = Cancel" giving you a summary of what to do next. LMB/RMB == Left/Right Mouse Button.

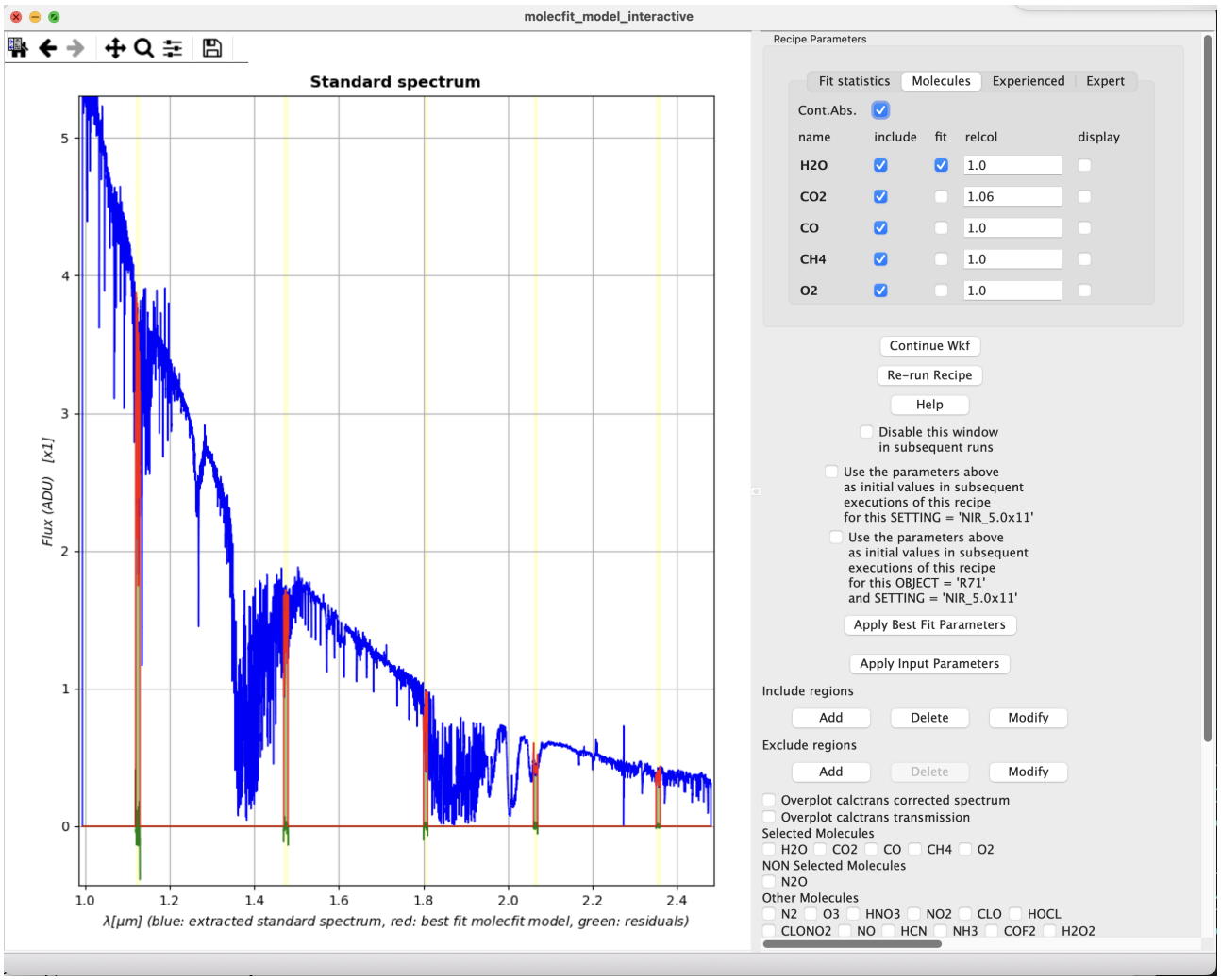


Figure 7.4: The interactive window displayed **after** the atmospheric fit (recipe: *molecfit_model*).

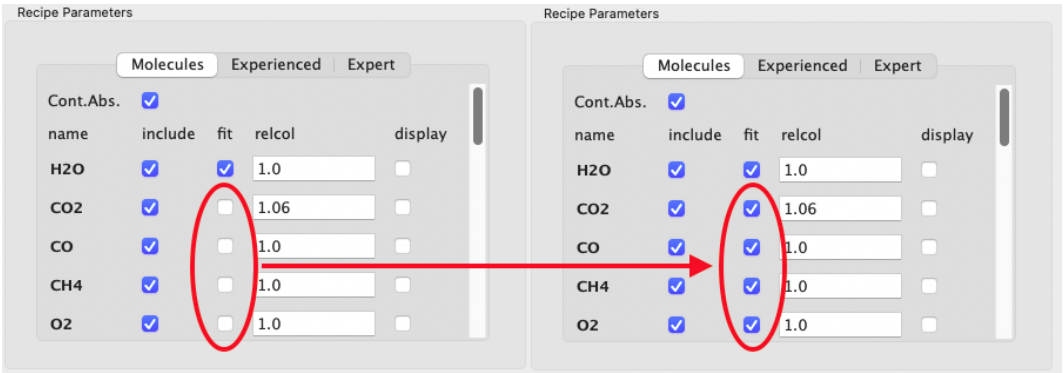


Figure 7.5: Select all relevant molecules in the XSHOOTER NIR wavelength range.

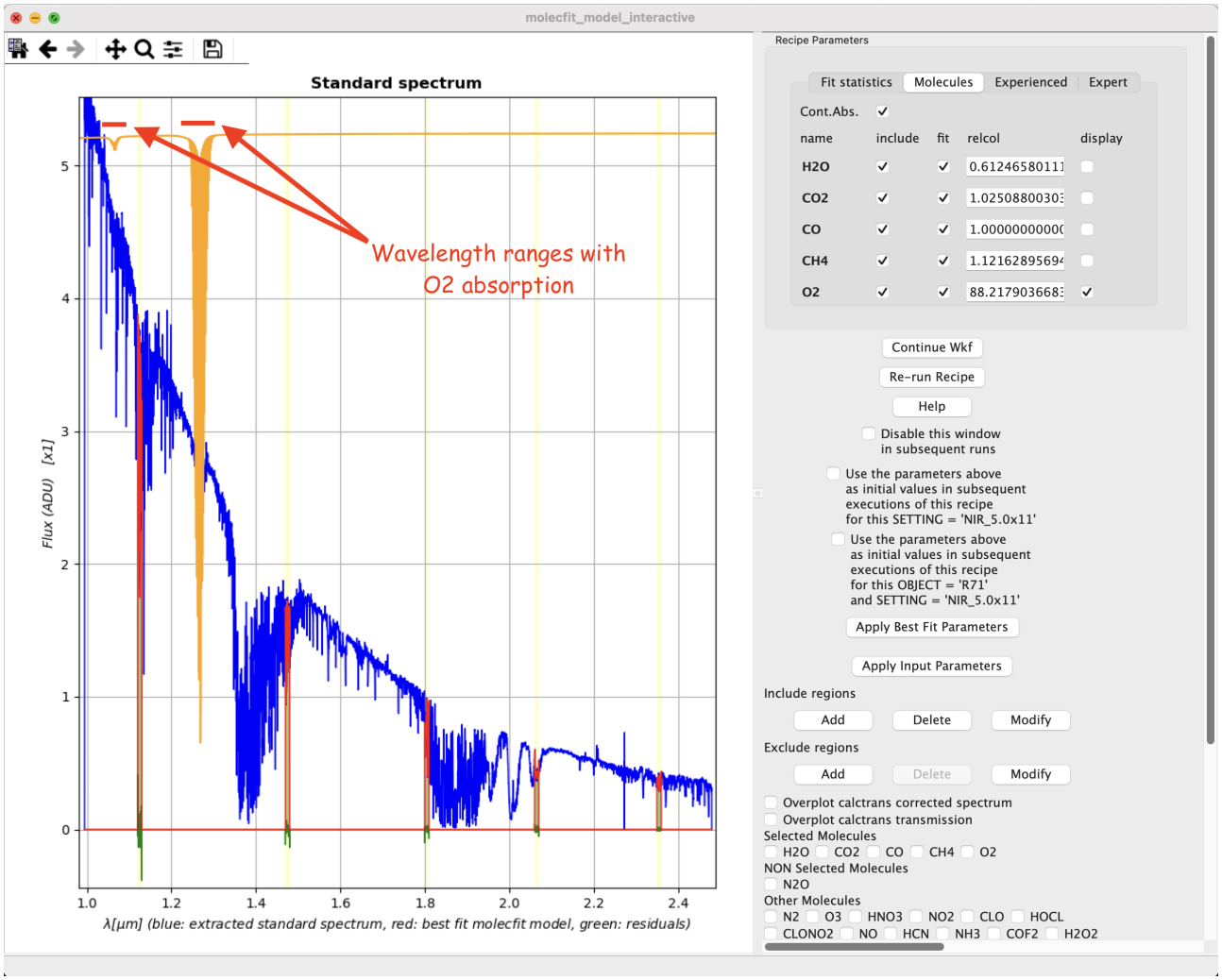


Figure 7.6: Displaying the O2 atlas.

- Position the mouse on the graph at xaxis value ∼1.23, click and hold the LMB and then drag the mouse to xaxis value ∼1.29 and then release the LMB.
- you can adjust the size of the region by click-dragging either edge of the region, or the position of the region by click-dragging from inside the region.[18]
- once you are happy with the region, click the 'Add' button again, and the newly defined region will be added to the current parameter value (for WAVE_INCLUDE).
- you can cancel tha Add process by right button clicking (on the plot) at any time (before clicking the 'Add' button for the 2nd time).
- you can delete and modify[19] existing WAVE_INCLUDE regions in a similar fashion, click the corresponding button, then select a region. To delete simple click the 'Delete' button again. To modify, adjust the size of the region by click-dragging either edge of the region, or the position of the region by click-dragging from inside the region and when happy with modifications, click the 'Modify' button again. And in both cases, RMB click anywhere in the plot to cancel.

After completing the above steps, you should see the newly defined fit wavelength region indicated in green. Unlike the ones of the input parameters, there is no yellow "background". See Figure 7.7.

With a fit region added to cover O2, re-run the recipe once again. In general, at each iteration it is probably advantageous to start from the Best Fit Parameter [BFP] values of the previous iteration since, as you close in on the ultimate model fit, it will presumably take less time to arrive at the new best fit from a starting point that is closer. But don't click the 'Apply Best Fit Parameters' button now, other wise you will loose the newly defined O2 fit region.

To proceed, click the **Re-run Recipe** button again to re-run the `molecfit_model` recipe with the new parameter settings.

The 'best_chi2' and 'reduced_chi2' statistics for the new fit including all five molecules and fit regions for all molecules should now be ∼30206672.7 & ∼16372.2 respectively[20] which rather seems like a more worse fit. Checking the new values of the BFPs, it seems that the REL_COL of O2 has hardly changed at all. So in this case starting from the BFPs, the fitting was unable to get out of the local minimum to a 'better' global minimum. So let's manually set the O2 REL_COL to 1.0 and try again (see Figure 7.8).

- Select the Molecules tab
- click in the relcol filed in the O2 row
- delete the current value and enter '1.0' – note the field background turns red indicating that 'un-commited' changes have been made
- type 'return'/'enter' to 'commit' the changed value

[18] On older installations, in particular with older versions of matplotlib (i.e. older than about version 2.0) – e.g. CentOS-7 – the behaviour is a little different due to the older implementation of the relevant matplotlib widget. As soon as you release the button, the region is defined and can not be adjusted. If you don't like the defined region, you unfortunately have to delete it and try again...

[19] Not available for older matplotlib versions.

[20] At this stage there could be 'significant' differences between your 'best_chi2' and 'reduced_chi2' statistics and the ones reported here depending on the exact definition of the O2 fit region you defined.

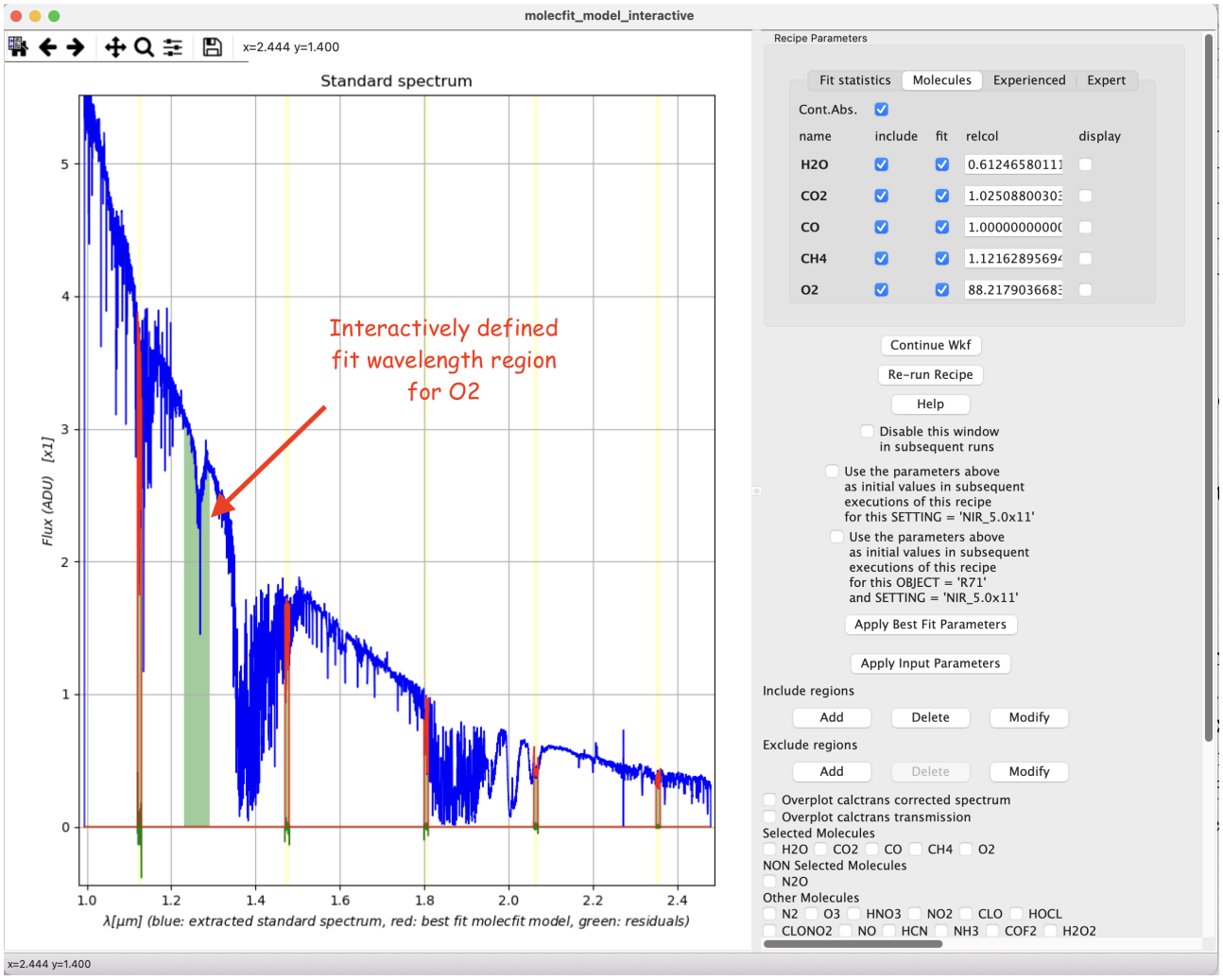


Figure 7.7: A fit region has been interactively added for O2.

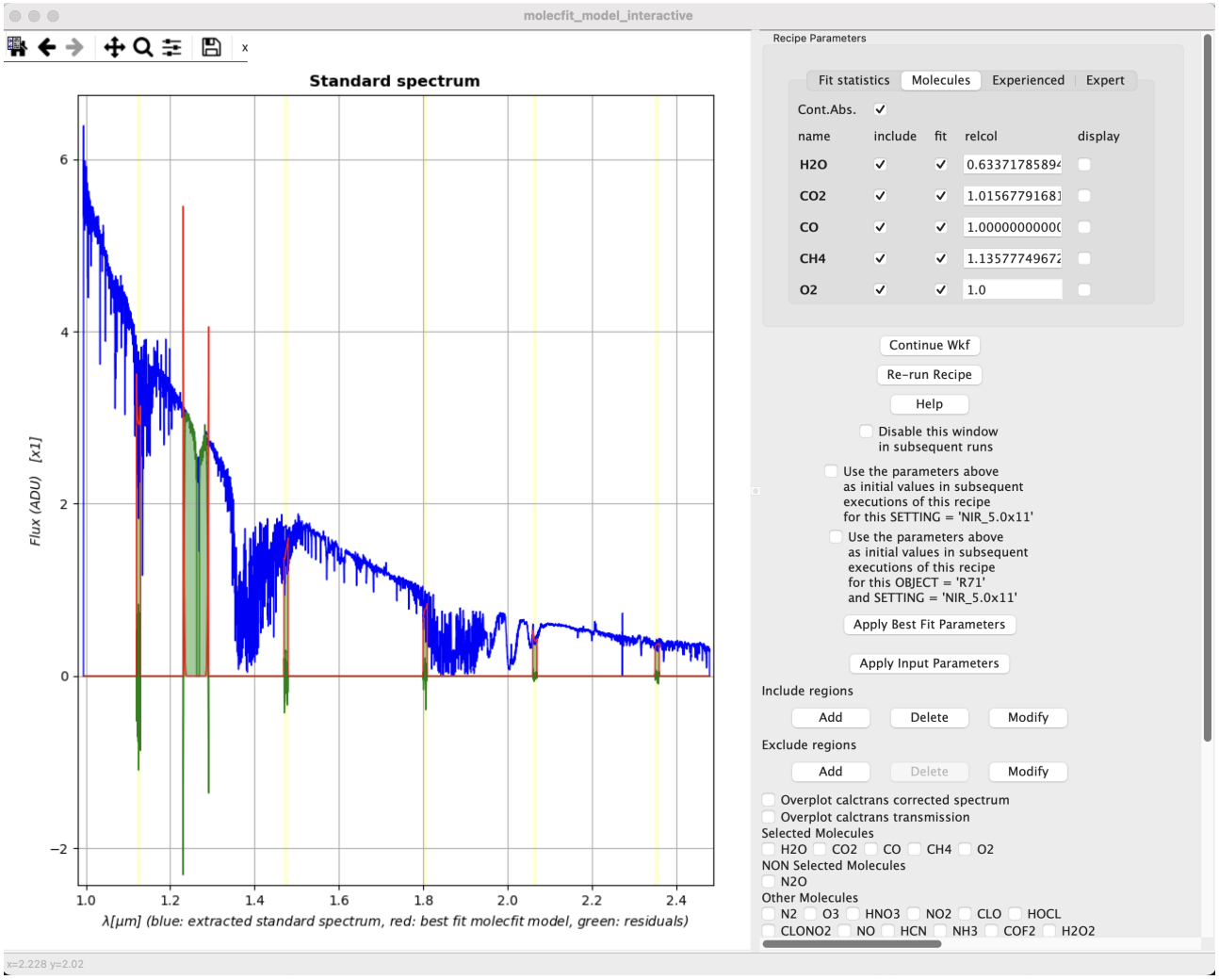


Figure 7.8: Direct adjustment of the O2 relcol value to escape the 'bad' local minimum.

To proceed, click the **Re-run Recipe** button again to re-run the `molecfit_model` recipe with the new parameter settings.

That helps! The 'best_chi2' and 'reduced_chi2' statistics for the new fit including all five molecules, fit regions for all molecules and a reasonable initial guess for the O2 REL_COL should now be ∼14655.2 & ∼7.9 respectively which seems like an improvement, plus in the Molecules tab, after applying the BFPs, all the relcol values look not unreasonable (near 1.0) – though the CO value of exactly 1.0 to 10dp at least looks a little suspicious... Turning on the CO atlas reveals that of the currently defined fit regions only the one at 2.35 to 2.36 μm includes CO lines, but those are quite badly contaminated by CH4 (turn on the CH4 atlas at the same time as the CO atlas). Extending that region to cover a larger number of the CO lines might help. Use the Include region Modify button to resize that region to 2.32 to 2.38 μm.

Meanwhile zooming in on the residuals of the O2 fit region, we can see some systematic deviations from noise, probably due to spectral features in the object. Turning on the O2 atlas (while zoomed) can can also see that the region we defined is rather too large. We should reduce the siae of the fit region and add an exclude region for the one feature that is probably still due to an object spectra feature, see Figure 7.9.

Once again, click the **Re-run Recipe** button to re-run the `molecfit_model` recipe with the new parameter settings.

Although it probably took significantly longer to compute than previous iterations, the 'best_chi2' and 'reduced_chi2' statistics for the new fit including all five molecules, improved fit regions for all molecules and starting from a more reasonable set of BFPs as first guess should now be ∼19207.6 & ∼9.6 respectively, which perhaps seems like a small step backward. Moreover in the Molecules tab, after applying the BFPs, all the relcol values still look not unreasonable (near 1.0), but the relcol for CO still seems stuck at 1.0 to (at least 10dp). That's a little bit the nature of molecfit, some trial and error, two steps forward, one step back. There's probably just no such thing as a perfect set of parameters which will yeild a perfect modelling for every observation of every object. But obvioulsy the more time you can invest, probably the better the model you can arrive at, within limits...

At this point we can check to see what the corrected spectrum would look like over the full wavelength range (not just the fit regions). Check the checkbox labelled 'Overplot calctrans corrected spectrum' (just below the exlude regions Add/Delete/Modify buttons). The fully corrected spectrum will be overplotted in grey, see Figure 7.10

7.2.2 Experienced/Expert user level tips for an improved fit.

New in ver. 4.3: The following tips can further help improve the quality of the fit. The parameters mentioned below are found in the Experienced or Expert Recipe Parameter tabs.

- **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.

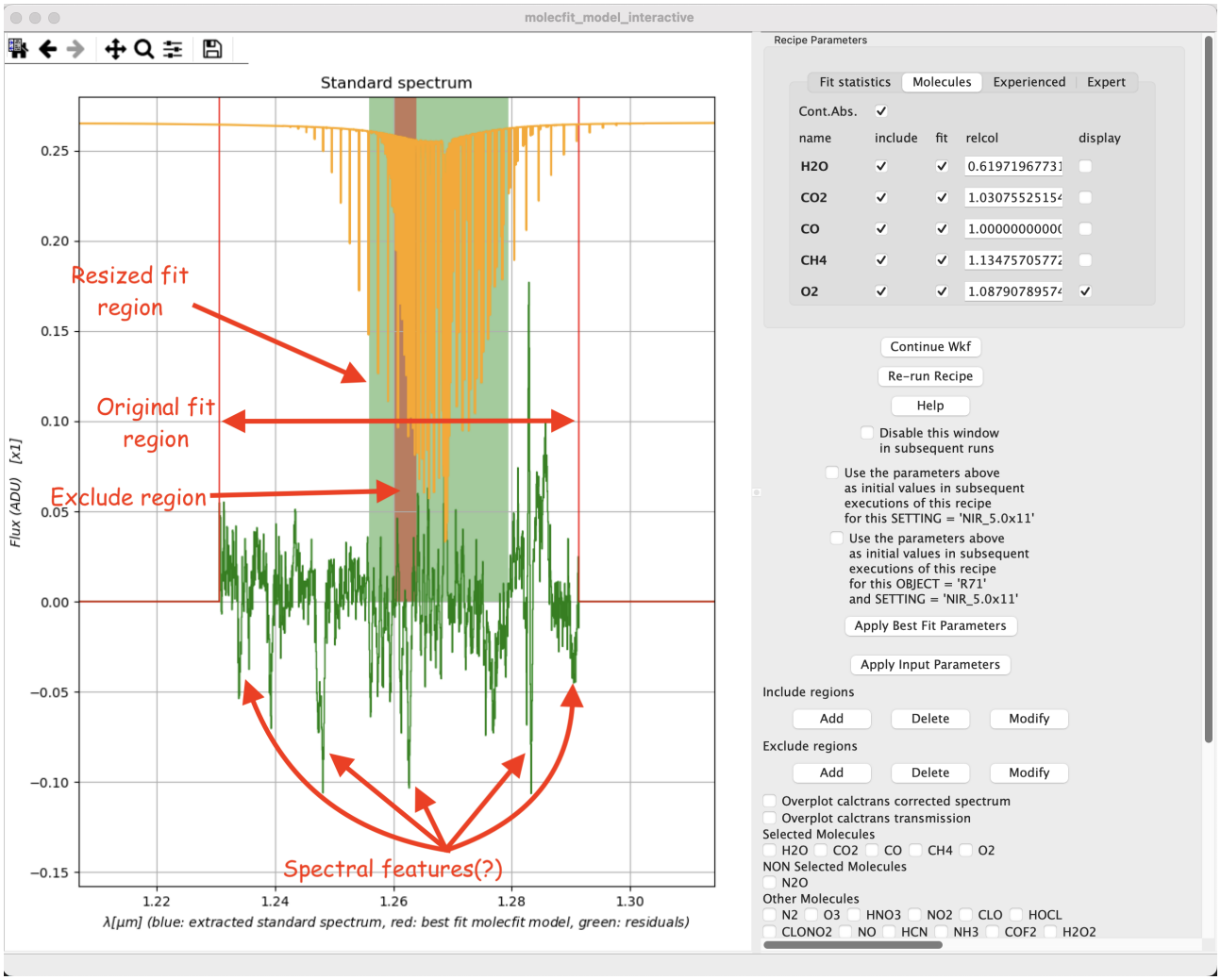


Figure 7.9: Modified O2 fit region with an exclude region.

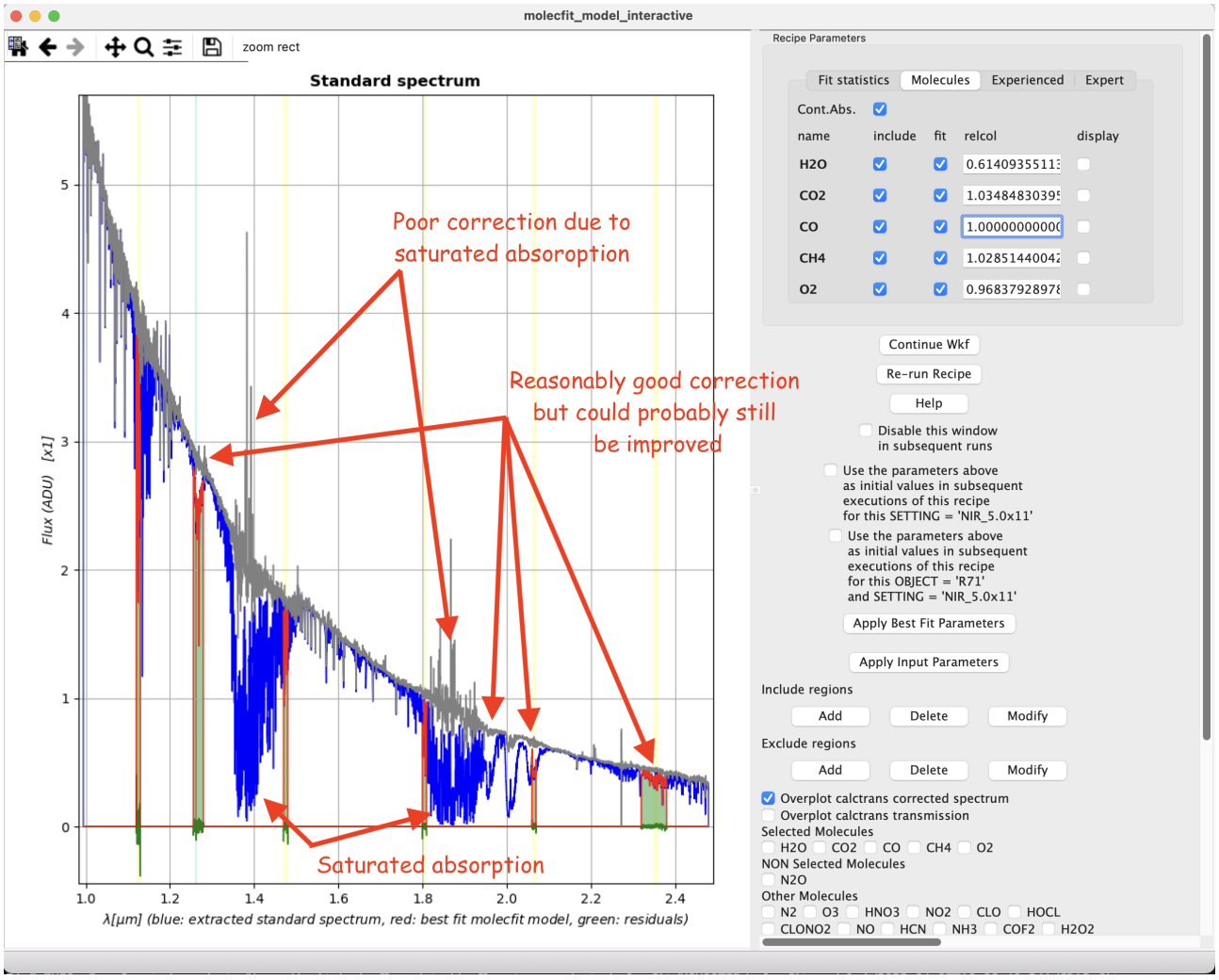


Figure 7.10: Overplotted corrected spectrum over full wavelength range (grey).

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 relevantr 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.

- **VISIR spectra**. VISIR spectra could be affected by fringing, i.e. an high-frequency modulation of the continuum level. This effect might have an impact when modeling the telluric features, therefore it is advisable to remove any fringing pattern from the VISIR spectra before running molecfit on it. Future releases of the VISIR pipeline will provide tools to remove fringes from the spectra.

7.2.3 Use the above parameters as initial values...

New in ver. 4.3: Once you have a 'good' set of parameters, you can save them for use as initial guesses of the parameters for future execution of the recipe. Because of the particular nature of molecfit – compared with other esoreflex workflows that in general deal only with data from a single given instrument – the molecfit workflow supports two 'flavours' of saving:

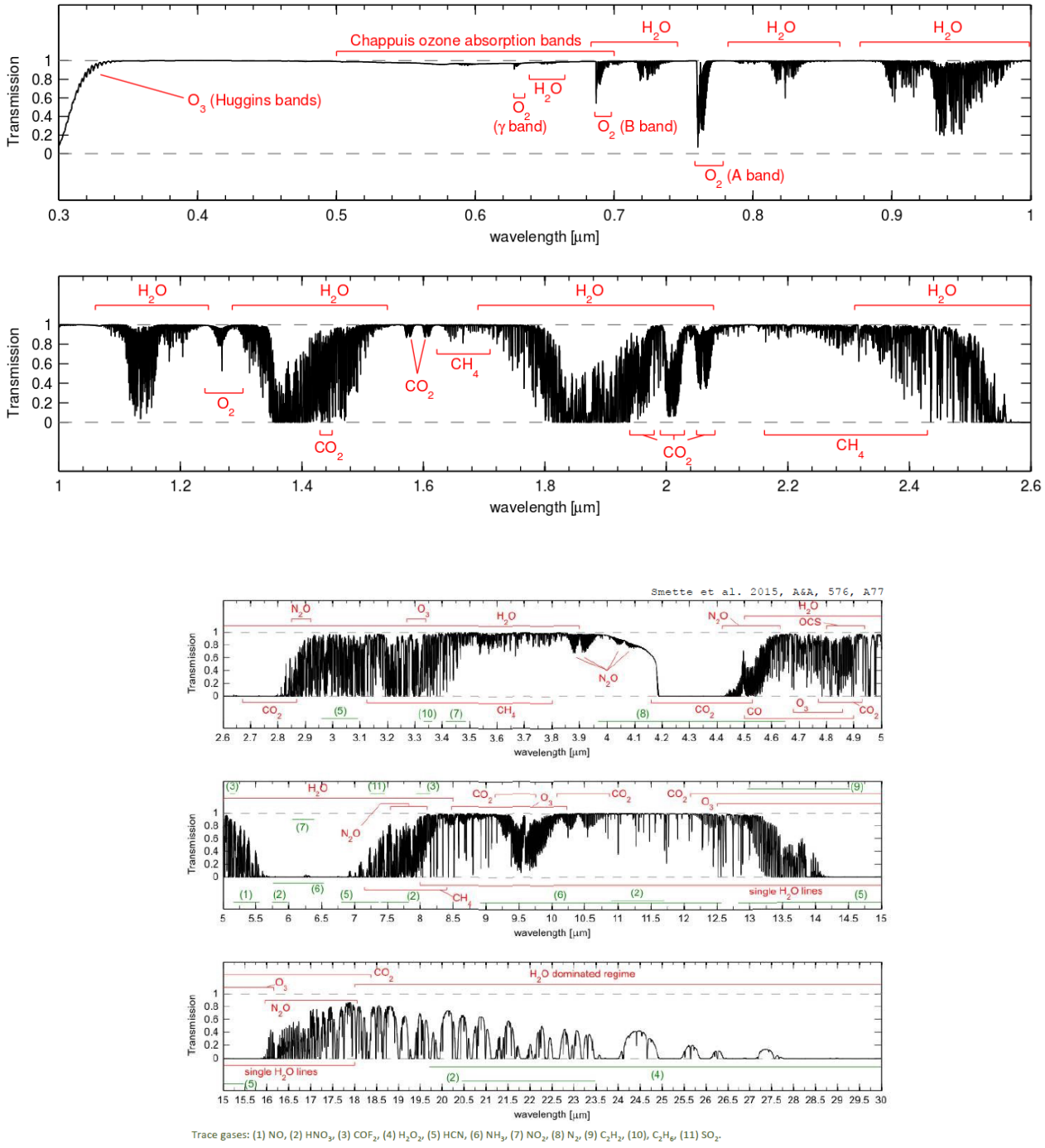


Figure 7.11: Synthetic absorption spectrum of the sky between 0.3 and 10 microns calculated with LBLRTM (resolution R= 10000) using the annual mean profile for Cerro Paranal (Noll et al. 2012), adapted from Smette et al. 2015, A&A, 576, 77. The eight main molecules O_2, O_3, H_2O, CO, CO_2 , CH_4, OCS, 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.

- **Instrument-setting**: save the current settings for future use with datasets of the same setting for the same instrument. The Instrument-setting is constructed from various header keywords, exactly which ones depends on the isntrument.
- **Object & Instrument-setting**: save the current settings for future use with datasets of the same objec **and** setting for the same instrument. The Instrument-setting is constructed as above while the object name is read from the 'HIERARCH ESO OBS TARG NAME' keyword if present otherwsie the 'OBJECT' keyword. Objects are matched by simple string match on the keyword value – nothing sophisticated such as matching coordinates...

An important difference between the molecfit and standard esoreflex implementations for saving parameters is that for the molecfit workflow parameter sets will be saved if the relevant checkbox is checked and either the **Re-run Recipe** or **Continue Wkf** buttons are pressed. For standard esoreflex implementations parameter sets are **only** saved when the **Continue Wkf** button is pressed. Note for both cases, no parameters are ever saved if the user closes the GUI window without clicking one of the **Re-run Recipe** or **Continue Wkf** buttons.

7.2.4 Apply ... Parameters buttons

New in ver. 4.3: Depending on what, if anything, has been done before, the setting of the main canvas parameter `GlobalReusePreviousExecution` and whether the `molecfit_model` GUI is being displayed before or after recipe execution, one or more 'Apply ... Parameters' buttons will be available, see Figure 7.12.

Clicking one of these buttons will set ALL parameters to the values from the corresponding parameter set. **Note: any manually set parameters will be 'lost'**.

When run before recipe execution, the GUI will always display an 'Apply Default Parameters' button. These are always the parameters derived by the workflow and python plugins for the instrument, instrument-setting and data-format.

When run after recipe execution, the GUI will always display an 'Apply Input Parameters' button. These are always the input parameters of that recipe execution.

If the main canvas parameter `GlobalReusePreviousExecution=true` then the 'Apply prev. successful exec. Parameters' button will be displayed in the `molecfit_model` GUI when displayed before recipe execution. **The initial values of the parameters when the GUI opens will be set to those values**.

If one or other (or both) of the 'Use the above parameters as initial values...' options has been previously used **and** the Instrument-setting or Instrument-setting **and** Object of the current dataset matches that of a previously processed dataset for which the option was used, then an 'Apply saved parameters for ...' button will be displayed.

7.3 Computation of atmospheric transmission: running `molecfit_calctrans`

Once the parameters of the atmosphere model have been fitted (either from a telluric standard or from a science frame), the workflow passes them to the `molecfit_calctrans` recipe to calculate the telluric correction over the full wavelength range of the input science spectrum.

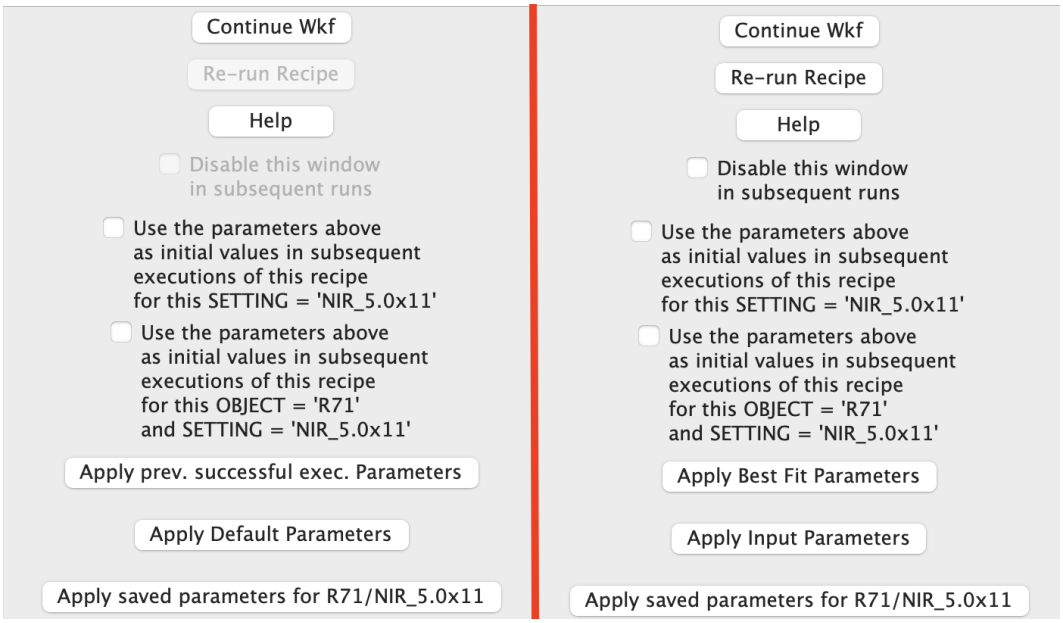


Figure 7.12: Apply Parameters options are slightly different depending on whether the `molecfit_model` GUI is being displayed before executing the recipe for the first time (left), or after (right). The 'Apply prev. successful exec. Parameters' button is only available before executing the recipe and only if the main canvas parameter `GlobalReusePreviousExecution=true`. The 'Apply Best Fit Parameters' button is only available after executing the recipe

The calculation 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 7.13.

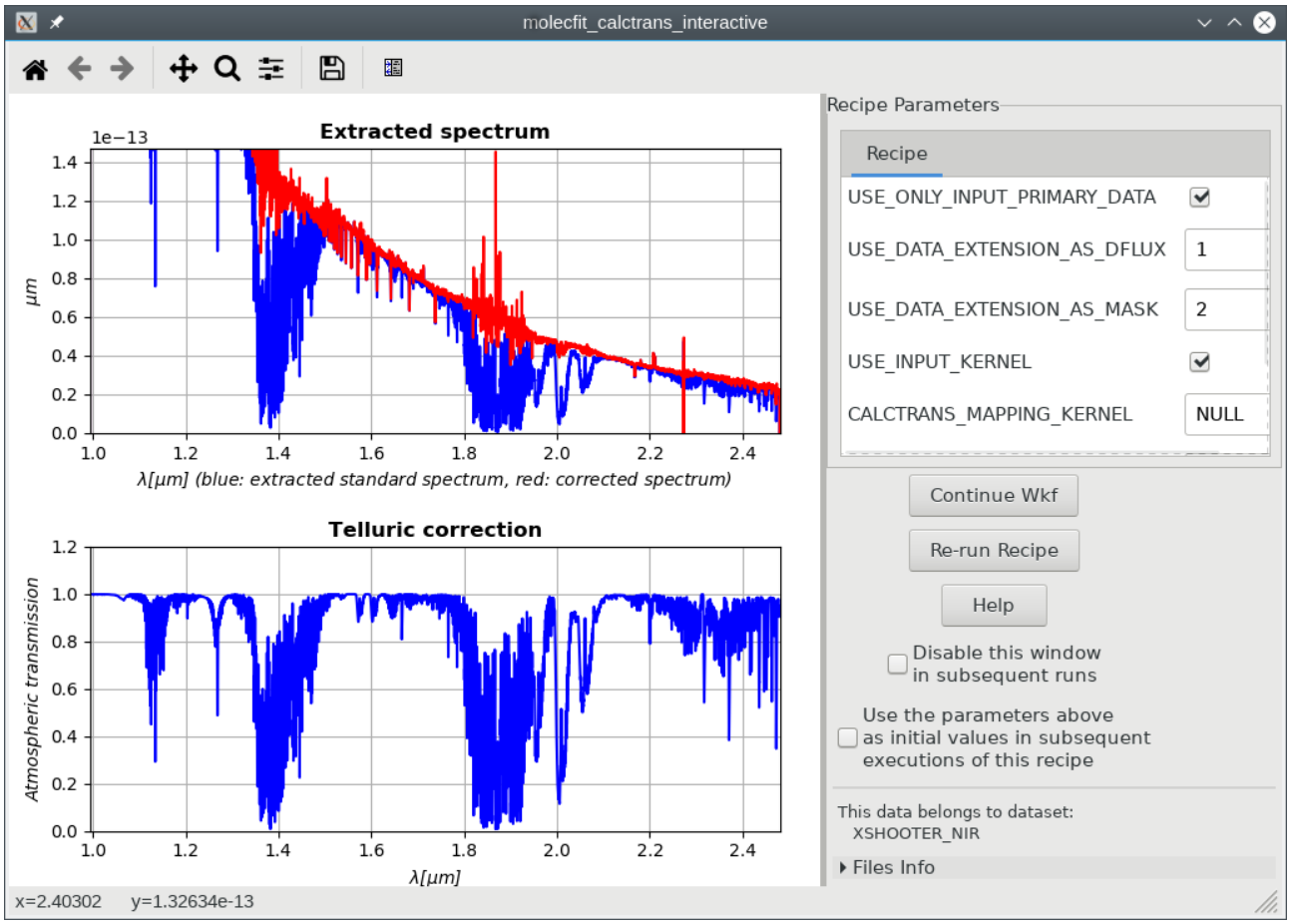


Figure 7.13: The interactive window for the recipe: *molecfit_calctrans*).

7.4 Correction for atmospheric transmission: running `molecfit_correct`

Once the full telluric corrections are computed by the calctrans step, the workflow applies them to the science spectra. This step is done by the actor `correct data` that is 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 7.14. 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

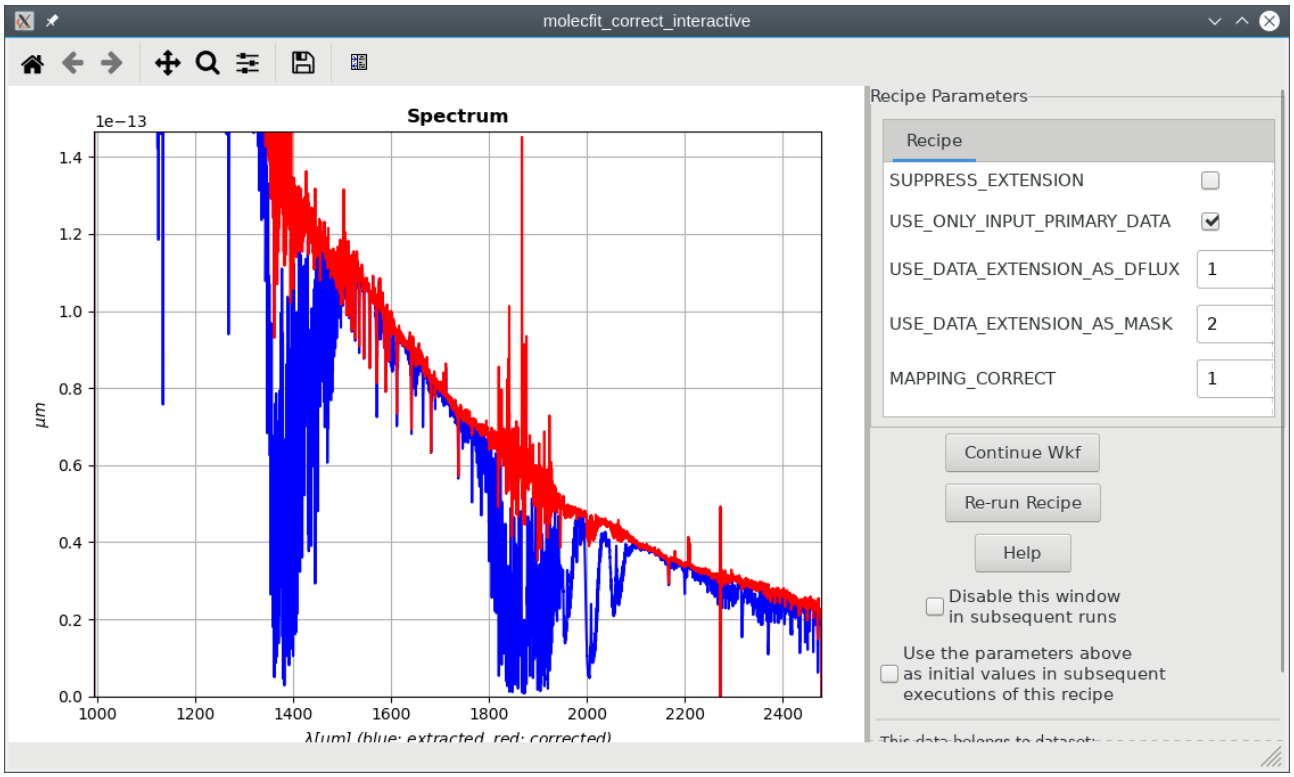


Figure 7.14: The interactive window for the `Telluric correction` (recipe: *molecfit_correct*).

8 Correcting other instrument data

New in ver. 4.3: Please note, this section (8) has not yet been updated for the improved `molecfit_model` GUI interface (point and click molecules and WAVE_INCLUDE/EXCLUDE region setting), so while the strategies outlined in the following are still valid, there are of course easier ways to apply them as of version 4.3, at least when it applies to adjusting molecules included/fitted and adjusting fit wavelengths.

In this section we provide instructions and tips on how to correct the data of the instruments (ESPRESSO, GIRAFFE, UVES, and VISIR) supported by the workflow or an arbitrary spectrum (not necessarily an ESO instrument). For convenience, we will refer to the demo data, but the concepts described here can be adapted to most of the cases.

Because it can't be stressed too much, we repeat here what was stated in section 7. As a general rule, it should be stressed that the automatically set parameters should always be considered as a *hopefully* useful first guess. In probably most, if not all real world use cases it will almost certainly be necessary to adjust and fine tune at least some of the molecfit_model parameters, in particular the wavelength regions to fit (WAVE_INCLUDE) and to carefully consider list of molecules (LIST_MOLEC) being fitted.

8.1 Correcting ESPRESSO data

8.1.1 Supported input file formats

ESPRESSO ADPs and IPPs are supported.

8.1.2 Optimizing

To correct ESPRESSO data, set the value of the global parameter 'INSTRUMENT' in the main canvas to ESPRESSO. To do so, double click with the mouse on the parameter, type ESPRESSO in the pop-up window and press 'OK'. If needed, adjust the SetUp directories to that they point so the desired locations and to the desired dataset(s). The default configuration will automatically point to the ESPRESSO demo data. The molecfit workflow can correct ESPRESSO data of the following categories:

- S1D_A
- S1D_FINAL_A
- S1D_FLUXCAL_A
- S1D_SKYSUB_A
- S1D_SKYSUB_FLUXCAL_A

The correction is done on the data, without association to telluric standards (not foreseen in the ESPRESSO night calibration plan).

Now, you can start the workflow by pressing the Play button. As for the XSHOOTER case, the workflow will automatically set the most commonly used recipe parameters for ESPRESSO. ESPRESSO data are set to the

barycentric velocity reference system, meaning that the telluric absorption line features are shifted with respect to their original wavelengths. This is taken into account by molecfit, by reading the velocity correction that was applied to the data from the header keyword `HIERARCH ESO QC BERV`. Extra-galactic sources (typically QSO) are not corrected for barycentric velocity by the ESPRESSO pipeline, therefore the value of `HIERARCH ESO QC BERV` should be 0.

In general, correction of ESPRESSO data is straightforward, as the wavelength range and the molecules present are always the same. If one inspects the molecfit best fit model and/or tries to change the default parameters, please keep in mind that, due to the very high resolution of the data, there could be many more target-related absorption features than telluric features (obviously, depending on the target).

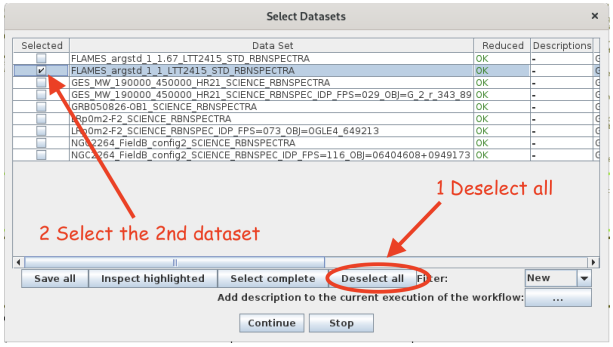


Figure 8.1: Selecting the 2nd GIRAFFE demo dataset.

8.2 Correcting GIRAFFE data

8.2.1 Supported input file formats

GIRAFFE ADPs and IPPs are supported. For IPPs the 2-D[21] RBNSPECTRA are 'reformated' into 1-D Binary Table input files by extracting the single fibre with the highest S/N, which is calculated using the DER_SNR[22] algorithm applied to the full spectral range of each fibre individually. More sophisticated algorithms maybe implemented in a future version.

8.2.2 Optimizing

To correct GIRAFFE data, set the value of the global parameter 'INSTRUMENT' in the main canvas to GIRAFFE. To do so, double click with the mouse on the parameter, type GIRAFFE in the pop-up window and press 'OK'. If needed, adjust the SetUp directories so that they point to the desired locations and to the desired dataset(s). The default configuration will automatically point to the GIRAFFE demo data.

Now, you can start the workflow by pressing the Play button. As for the XSHOOTER case, the workflow will automatically set the most commonly used recipe parameters for GIRAFFE. The wavelength intervals and the molecules to fit are automatically set by the internal auto-molecule algorithm (see appendix A), depending on the wavelength range of the data. Obviously this can, and in general *should* be overridden by the user in the interactive window for the current dataset since the auto-molecule set wavelength ranges are in general sub-optimal. If you want the new recipe parameters to be used for all the datasets, then disable the parameter initialization before starting the workflow (set the global parameter in the canvas 'ParameterInitialization' to false).

For the present purposes, in the DataSet Chooser window, select just the 2nd dataset for processing (click the "Deselect all" button, and then click the checkbox to the left of the 2nd dataset), see Figure 8.1.

Once the `molecfit_model` interactive GUI window appears you should see something that looks like Figure 8.2.

[21] 2-D Image HDUs with spectra for each fibre in the columns of the image.
[22] *http://www.stecf.org/software/ASTROsoft/DER_SNR/*

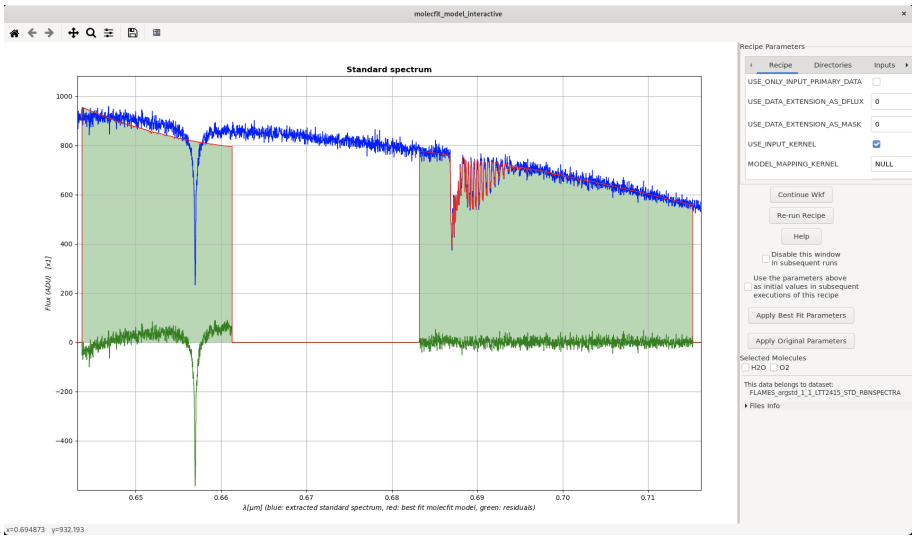


Figure 8.2: The result of the modeling step on the 2nd GIRAFFE demo dataset with automatic parameter initialization. The fit to the continuum in the shorter-wavelength fit region (on the left) is sub-optimal due to the strong absorption feature from the object at $\sim$ 0.6568 micron.

The fit is quite reasonable, but the fitting range on the left clearly includes a strong absorption line from the object.

Note that over the visible wavelength range, only two molecules (H2O and O2) have significant[23]) absorption features, for this data range both molecules are relevant. You can toggle on/off displaying an atlas of each molcule from the panel on the right (you may need to resize or scroll the panel depending on the size of the GUI window on your monitor.)

So we want to re-run the recipe, with better parameters. In most cases it is a good idea to apply the parameters of the current fit (click the "Apply Best Fit Parameters" button) so that the fitting will start from a point closer to the optimum, and will thus (usually) take less time to converge than if we start from the default aut-set parameters. Doing so now should set the `REL_COL` parameters to:

```
REL_COL = 0.332357,0.982580
```

Now the fit can certainly be improved by excluding the absorption line from the fitting region (e.g. shortening the first fit region to from 0.643730,0.661322 to say 0.643730,0.653626). But turning on the molecule atlas displays (see figure 8.3) we can see that using just a part of the fitting region on the right that includes absorption lines from both H2O and O2 should be enough to provide a reliable fit for both molecules.

If one changes the wavelength regions to fit to:

```
WAVE_INCLUDE = 0.6868,0.7035
```

then a much better fit can be achieved, as shown in Figure 8.4.

[23]The GIRAFFE `auto-molecule` file included in the molecfit datastaic package has been prepared with a threshold of 2%, see section A.

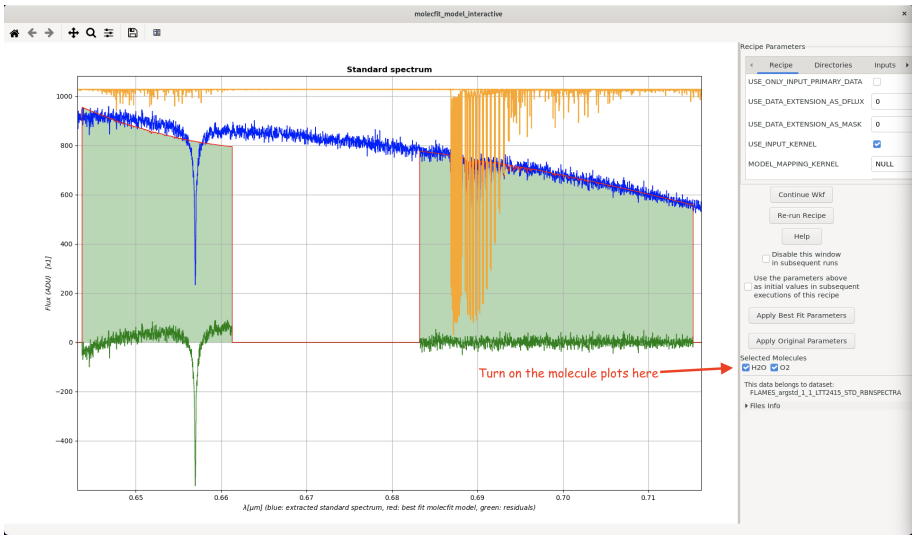


Figure 8.3: With the H2O and O2 molecule atlases displayed, we can see that by fitting a sub region of the default longer-wavelength fit region (on the right) we should be able to get a reliable solution and save significant computation time.

The resuling best for `REL_COL` figures are now:

```
REL_COL = 0.611221,0.935142
```

The significant change in the `REL_COL` for H2O presumably results mostly from no longer including the absortion feature from the object in the fit.

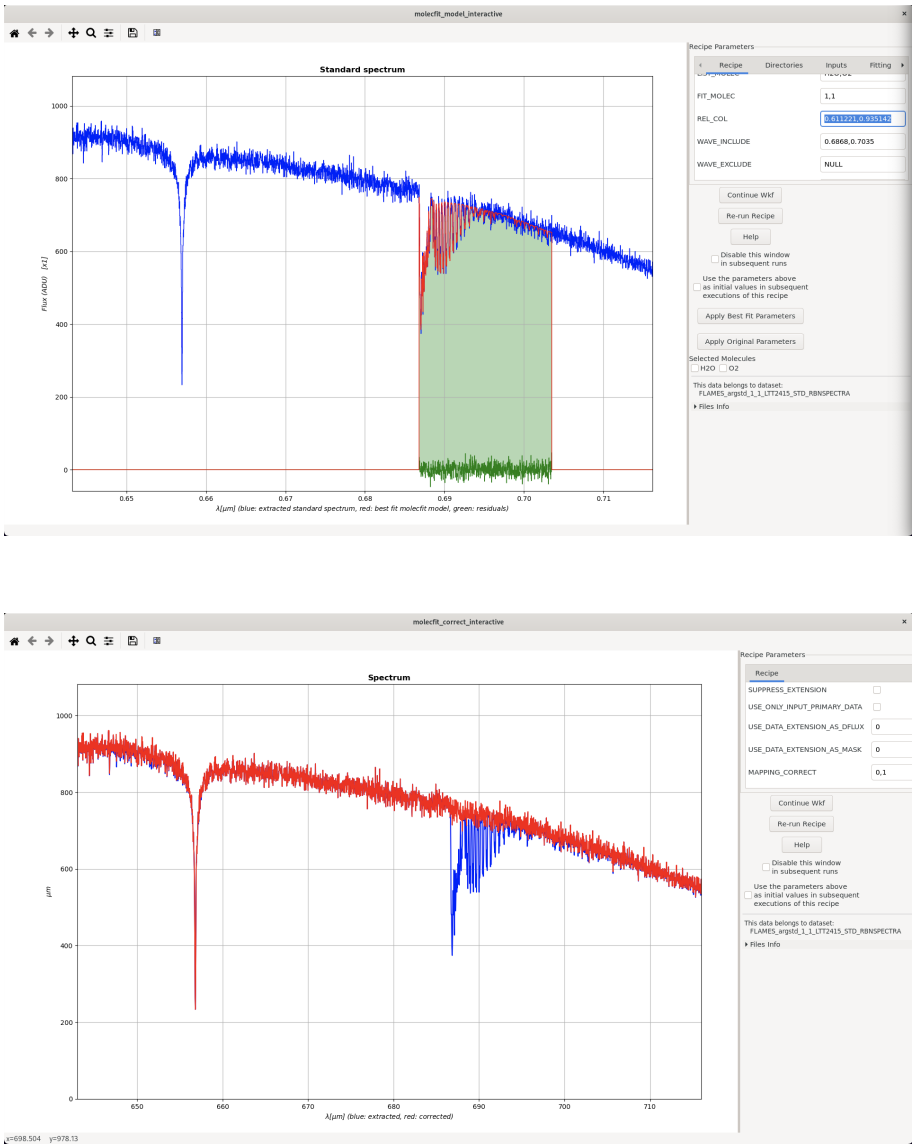


Figure 8.4: Upper panel, the result of the modeling step on the 2nd GIRAFFE demo dataset with ‘optimized’ WAVE_INCLUDE. Lower panel, final corrected spectrum.

8.3 Correcting UVES data

8.3.1 Supported input file formats

UVES ADPs and IPPs are supported, but the workflow only supports RED arm data, BLUE arm data are ignored, this (may) change in a future relase.

8.3.2 Optimizing

To correct UVES data, set the value of the global parameter 'INSTRUMENT' in the main canvas to UVES. To do so, double click with the mouse on the parameter, type UVES in the pop-up window and press 'OK'. If needed, adjust the SetUp directories so that they point to the desired locations and to the desired dataset(s). The default configuration will automatically point to the UVES demo data.

Now, you can start the workflow by pressing the Play button. As for the XSHOOTER case, the workflow will automatically set the most commonly used recipe parameters for UVES. The wavelength intervals and the molecules to fit are automatically set by the internal auto-molecule algorithm (see appendix A), depending on the wavelength range of the data, except for WL-setting=860 data for which the there are static calibrations specifying the wavelength regions to fit and molecules to fit. Of course, the wavelength regions to fit set by auto-molecule can, and in general *should*, be optimised by the user in the molecfit_model interactive window for each dataset since the auto-molecule set wavelength ranges are in general sub-optimal. Even for the WL-setting=860 data the user should take time to optimise the fitting ranges for the specific data being processed.

If you want the new recipe parameters to be used for all the datasets, then disable the parameter initialization before starting the workflow (set the global parameter in the canvas 'ParameterInitialization' to false).

For tutorial purposes, the workflow automatic parameterization does not provide a good fit for the first dataset of the datademo package ADP.2020-06-08T15:07:14.471 (or indeed any of the demo datasets). The results of actor Model the atmosphere is shown in Figure 8.5. If one changes the wavelength regions to fit to:

```
WAVE_INCLUDE = 0.586,0.60,0.625,0.64,0.645,0.653
```

then a much better fit can be achieved, as shown in the upper pannel of Figure 8.6. Continuing with the workflow ("Continue WKF" button) however shows in the calctrans display that infact the Relative Column density of O3 has been largely over estimated. How did that happen?

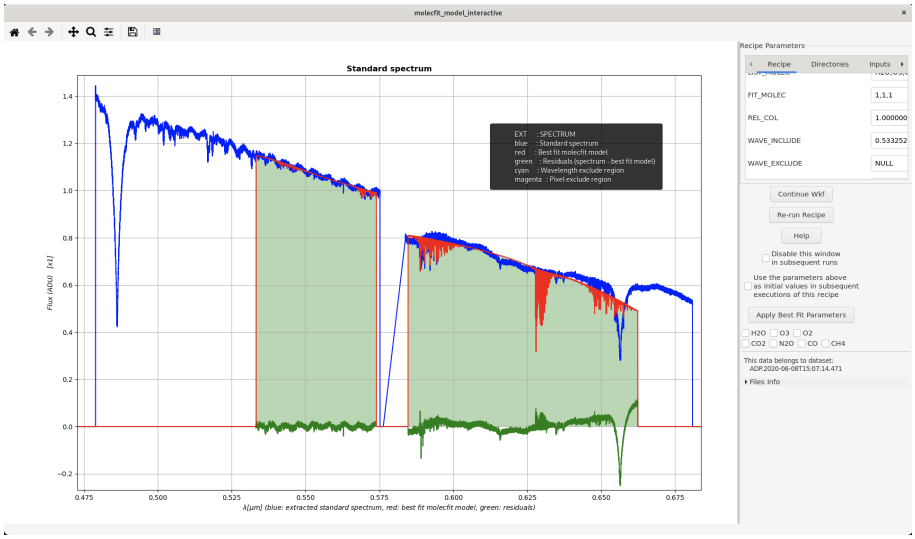


Figure 8.5: The result of the modeling step on the demo dataset ADP.2020-06-08T15:07:14.471 with automatic parameter initialization. The fit is not great, the residuals (green line), especially for the upper chip, show significant deviations from noise about zero; the fit to the continuum is clearly not good and a number of stellar absorption lines are included in the fitting range.

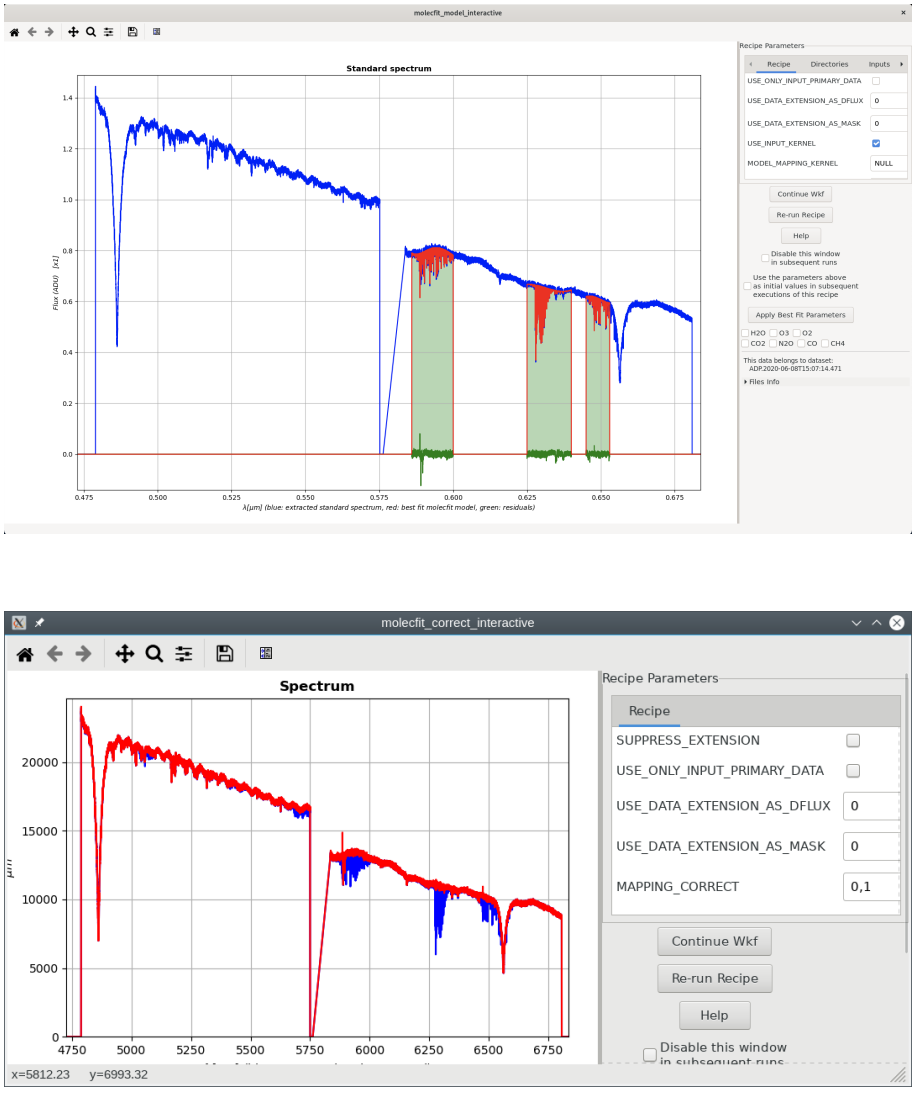


Figure 8.6: Upper panel, the result of the modeling step on the demo dataset ADP.2020-06-08T15:07:14.471 with WAVE_INCLUDE (other fitting parameters as set by the automatic parameter initialization). Lower panel, final corrected spectrum. Also few telluric absorption outside the fitted region are nicely cleaned.

8.4 Correcting VISIR data

8.4.1 Supported input file formats

VISIR IPPs are supported (ADPs are not currently available for VISIR data).

8.4.2 Optimizing

To correct VISIR data, set the value of the global parameter 'INSTRUMENT' in the main canvas to VISIR. To do so, double click with the mouse on the parameter, type VISIR in the pop-up window and press 'OK'. If needed, adjust the SetUp directories to that they point to the desired locations and to the desired dataset(s). The default configuration will automatically point to the VISIR demo data.

Now, you can start the workflow by pressing the Play button. As for the XSHOOTER case, the workflow will automatically set the most commonly used recipe parameters for VISIR. The wavelength intervals and the molecules to fit are automatically set by the internal auto-molecule algorithm (see appendix A), depending on the wavelength range of the data. Obviously this can, and in general *should* be overridden by the user in the interactive window for the current dataset since the auto-molecule set wavelength ranges are in general sub-optimal. If you want the new recipe parameters to be used for all the datasets, then disable the parameter initialization before starting the workflow (set the global parameter in the canvas 'ParameterInitialization' to false).

The VISIR data take long time to fit. For example, the first dataset could take up to 1.5 hours on a laptop with 8GB RAM, 4 CPUs, Intel i5 2.30GHz). This is because the full wavelength range is fitted (divided into 5 regions) and all the molecules that contribute in that region are selected by the automatic selection procedure. To speed up the process one can decrease the wavelength range and the number of molecules to fit. qqIn order to better understand where a molecule contributes and how much this contribution is, we recommend to display them by using the dedicated toggle buttons in the interactive windows.

The telluric correction of VISIR data is complicated by the presence of systematic modulations of the continuum, due to non-optimal response functions in the data reduction. These modulations mimic broad telluric absorptions, and make the distinction between absorption features and continuum fluctuactions difficult. The very short wavelength range of VISIR data makes this degeneracy even more complex to solve. This problem will be solved in future VISIR pipeline releases. However, on some occasions, the empirical approach of dividing the science data by a continuum-normalized spectrum of a telluric standard (if available) might be the preferred approach.

8.5 Correcting XSHOOTER data

8.5.1 Supported input file formats

XSHOOTER ADPs and IPPs are supported, but the workflow currently does not support IFU products.

8.5.2 Optimizing

For demo purposes, the static calibrations for XSHOOTER only fit for H2O. Although other molecules are included, their contributions are all held fixed with REL_COL values of 1.0. Obviously for a more meaningful fit, these other molecules should be solved for too.

To illustrate in detail. Let's re-process the first XSHOOTER dataset. Set the value of the global parameter 'INSTRUMENT' in the main canvas to XSHOOTER (this is the default value so if you haven't changed it previously it will already be set to XSHOOTER and there is nothing to change). To do so, double click with the mouse on the parameter, type XSHOOTER in the pop-up window and press 'OK'. If needed, adjust the SetUp directories so that they point to the desired locations and to the desired dataset(s). The default configuration will automatically point to the XSHOOTER demo data.

Now, you can start the workflow by pressing the Play button. As described in the quickstart section, the workflow will automatically set the most commonly used recipe parameters for XSHOOTER. The wavelength intervals and the molecules to fit are set by static calibrations included in the molecfit calibrations package. the internal auto-molecule algorithm (see appendix A), depending on the wavelength range of the data, except for WL-setting=860 data for which the there are static calibrations specifying the wavelength regions to fit and molecules to fit. Of course, the the internal auto-molecule algorithm (see appendix A), depending on the wavelength range of the data, except for WL-setting=860 data for which the there are static calibrations specifying the wavelength regions to fit and molecules to fit. XSHOOTER is the one case where a certain amount of effort has gone into defining the wavelength regions to fit, but still the user should take time to optimise the fitting ranges for the specific data being processed.

Processing with the default parameters results in a perfectly reasonable looking fit, the residuals all look comparable with the spectrum noise close to the respective fit regions, see Figure 8.7.

Before making any changes, it is worth clicking the 'Apply Best Fit Parameters' button in order to restart from the current solution. Once done the `REL_COL` parameter will be set to:

```
REL_COL = 0.845532,1.060000,1.000000,1.000000,1.000000
```

Now the first improvement is to activate fitting for all the molecules. To do so, set the `FIT_MOLEC` parameter in the 'Recipe' tab:

```
FIT_MOLEC = 1,1,1,1,1
```

Once done, then click the 'Re-run Recipe' button.

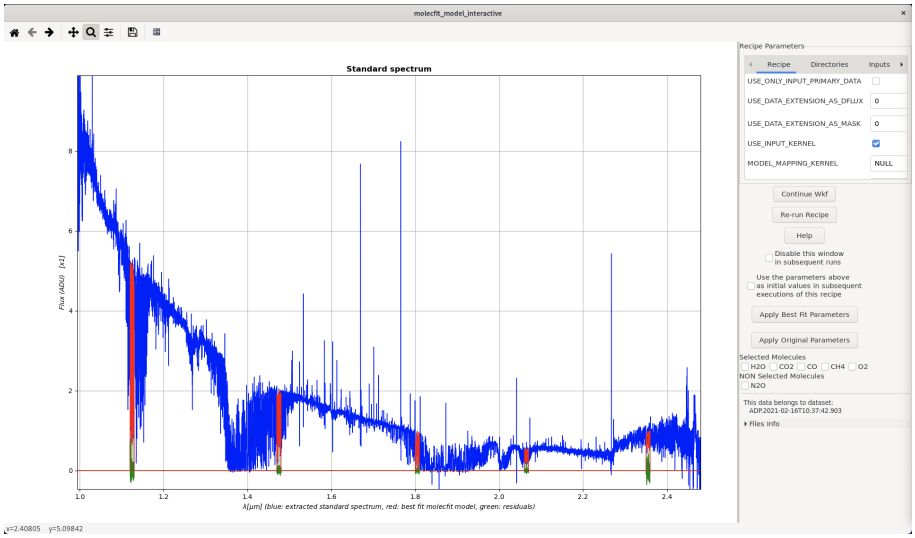


Figure 8.7: The result of the modeling step on the demo dataset ADP.2021-02-16T10:37:42.903 with automatic parameter initialization. The fit already looks pretty good, the residuals (green line), all look comparable with the spectrum noise close to the respective fit regions.

Once the recipe has completed and the GUI window reappears, although it is difficult to see by eye and from memory any significant change (or any change at all) in the quality of the fit (see Figure 8.8), after applying the best fit parameters once agian, we can see that some of the fitted REL_COL values have changed significantly.

```
REL_COL = 0.843000,1.053864,10.306833,0.393716,0.000010
```

Hmmmmm! A `REL_COL` of 10.3 for CO seems a little unlikely. What's gone wrong there?

If you turn on the atlas for CO (see Figure 8.9), you see that the only fitting region applicable to CO is the one from `2.35,2.36`. If you also turn on the atlases for H2O and CH4 (in turn), you also see that those other two molecules have dominating absorptions in that wavelength region. So unfortunately it looks like it will be rather difficult to make a meaningful fit to CO in this wavelength region. So it is thus probably better to set `FIT_MOLEC` for CO back to 0, and `REL_COL` for CO back to 1.0 i.e.:

```
FIT_MOLEC = 1,1,0,1,1
REL_COL = 0.843000,1.053864,1.0,0.393716,0.000010
```

Now what about O2? A `REL_COL` of 0.000010 also seems a little unlikely. Turning on the atlas for O2 (see Figure 8.10) it becomes clear what the problem is, there is no region defined that includes absortion lines for O2. Checking the atlases of all the other molecules shows that a fitting region around 1.269 μm, say `1.268,1.271`, should target the O2 line nicely. So now set `WAVE_INCLUDE` as follows:

```
WAVE_INCLUDE = 1.12,1.13,1.268,1.271,1.47,1.48,1.80,1.81,2.06,2.07,2.35,2.36
```

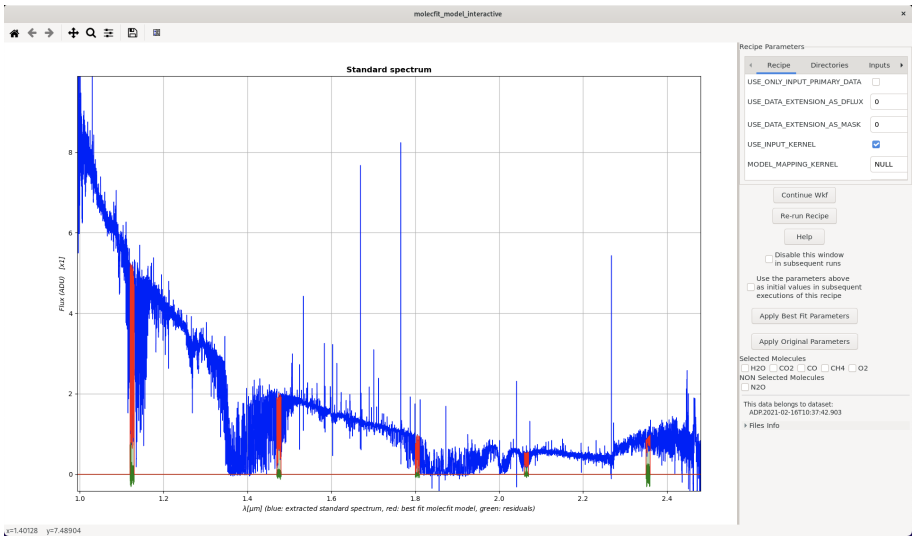


Figure 8.8: The result of the modeling step on the demo dataset ADP.2021-02-16T10:37:42.903 with with fitting turned on for all 5 selected molecules.

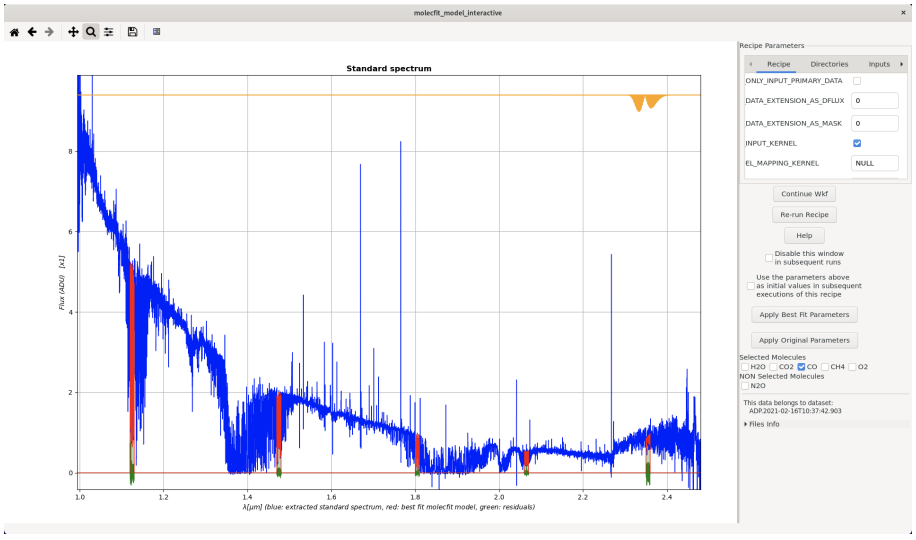


Figure 8.9: The result of the modeling step on the demo dataset ADP.2021-02-16T10:37:42.903 with with fitting turned on for all 5 selected molecules with the atlas turned on for CO.

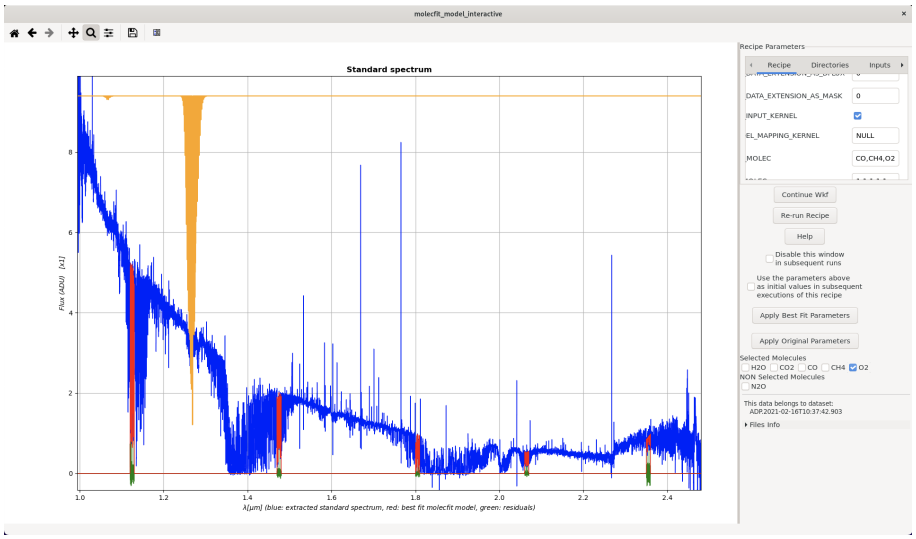


Figure 8.10: The result of the modeling step on the demo dataset ADP.2021-02-16T10:37:42.903 with with fitting turned on for all 5 selected molecules with the atlas turned on for O2.

And then once again click the 'Re-run Recipe' button.

The new fitted `REL_COL` values look a bit more sensible.

```
REL_COL = 0.843512,1.054379,1.0,0.433605,1.008141
```

But let's make sure of the rather low `REL_COL` value for CH4. There's a fairly strong CH4 feature with little contamination from other molecules at about 1.666 μm So now insert 1.665,1.667 in between 1.48 and 1.80 in the `WAVE_INCLUDE` setting[24]. Re-running the recipe with this additional fitting region specifically targeting CH4 results in a ∼50% increase in the CO4 `REL_COL` value, with only marginal change in the values for the other molecules.

```
REL_COL = 0.844080,1.055699,1.0,0.678286,1.025405
```

Continuing to improve the fit, and adding in fitting for N2O is now left as an exercise for the reader.

[24] `1.12,1.13,1.268,1.271,1.47,1.48,1.665,1.667,1.80,1.81,2.06,2.07,2.35,2.36`

8.6 Correcting ANY set of data

8.6.1 Supported input file formats

Input files *must* be in one of the two intrinsically supported molecfit input file formats, see section 6.4.1.

8.6.2 Optimizing

To correct data for instruments for which the automatic parameter set-up is not supported, set the value of the global parameter 'INSTRUMENT' in the main canvas to ANY. This setting recognizes some ESO instruments. It runs the model fit and the atmospheric transmission correction on the file directly, without associating any telluric standard star.

To correct any spectrum, even from non-ESO instrument, one has to prepare the input files to fulfill some basic requirements.

- Make sure the input file is a 1D spectrum. Supported files are IMAGE and BINARY table (both single-row or multi-row data format). Datacubes or long 2D long-slit spectra are not supported. The data extension can be either in the primary or first extension (recommended).
- For non-ESO instruments or for ESO instruments with a category[25] not recognized by the workflow, add or edit the header keyword `INSTRUME` in the primary header and set its value to `ANY`. This step is mandatory for non-ESO instruments or for those ESO instruments with a category not recognized by the workflow.
- **New in ver. 4.3:** The primary header of the file must also contain the `MJD-OBS` keyword.

New in ver. 4.3: The following `molecfit_model` canvas parameters should be set to the name of the keyword in the primary header of each file containing the appropriate value, or else these canvas parameters can be set to `NONE`, and the corresponding canvas parameter without the `_KEYWORD` suffix can be set to the actual value of that property appropriate for the (single) file to be processed – in this case one evidently then needs to edit the canvas for each input file.

```
ELEVATION_KEYWORD
LATITUDE_KEYWORD
LONGITUDE_KEYWORD
MIRROR_TEMPERATURE_KEYWORD
OBSERVING_DATE_KEYWORD
PIX_SCALE_KEYWORD
PRESSURE_KEYWORD
RELATIVE_HUMIDITY_KEYWORD
SLIT_WIDTH_KEYWORD
TELESCOPE_ANGLE_KEYWORD
TEMPERATURE_KEYWORD
UTC_KEYWORD
```

[25] Usually read from the Primary header keyword `PRODCATG` or `HIERARCH ESO PRO CATG`.

New in ver. 4.3: The easiest way to get help about these parameters (e.g. which units are required) is to create a config file for the `molecfit_model` recipe[26].

If the data are taken from a telescope for which GDAS atmospheric profile is available (see list here: https://ftp.eso.org/pub/dfs/pipelines/skytools/molecfit/Observatories.txt), then the recipes automatically downloads the corresponding profile; otherwise one has to create a suitable profile (please consult the molecfit user manual in this case).

Before starting the workflow, it is recommended to switch off the parameter initialization (set the variable **ParameterInitialization** to 'false' in the main canvas). Then it is also recommended to set the recipe parameters according to the input data (e.g., conversion of the wavelength to micron, name of the table columns to consider, wavelength ranges and molecules to fit).

The most convenient way to set the recipe parameters is

- double click on the desired interactive actors. A configuration window listing the parameters will appear. Note: the reflex parameter names are as the recipe parameter names, but with the `_INIT` suffix.
- Change the desired parameter and press commit. The configuration window will close.

If a parameter is not set correctly, the corresponding recipe can fail. Press continue to load the interactive window and change the recipe parameters.

8.6.3 Recipe parameters

The user has to take care of setting all relevant recipe parameters, depending on the data properties such as units and wavelength range. A full list of recipe parameters can be obtained from the molecfit pipeline manual. Here, we provide instructions only for the less intuitive parameters.

- If the data are stored in the primary extension:
  - Recipe `molecfit_model`
    `-MODEL_MAPPING_KERNEL = NULL`
    `-USE_ONLY_INPUT_PRIMARY_DATA = TRUE`
  - Recipe `molecfit_calctrans`
    `-USE_ONLY_INPUT_PRIMARY_DATA = TRUE`
    `-CALCTRANS_MAPPING_ CONVOLVE = NULL`
    `-MAPPING_CONVOLVE = "1"`
    `-MAPPING_ATMOSPHERE = "1"`
  - Recipe `molecfit_correct`
    `-USE_ONLY_INPUT_PRIMARY_DATA = TRUE`
    `-MAPPING_CORRECT = "1"`
- If the data are stored in the data extension (i.e., the primary extension contains only the header)

[26] `esorex -create-config=model.rc molecfit_model`

- Recipe `molecfit_model`
  `-MODEL_MAPPING_KERNEL = NULL`
  `-USE_ONLY_INPUT_PRIMARY_DATA = FALSE`
- Recipe `molecfit_calctrans`
  `-USE_ONLY_INPUT_PRIMARY_DATA = FALSE`
  `-CALCTRANS_MAPPING_ CONVOLVE = NULL`
  `-MAPPING_CONVOLVE = "1,1"`
  `-MAPPING_ATMOSPHERE = "0,1"`
- Recipe `molecfit_correct`
  `-USE_ONLY_INPUT_PRIMARY_DATA = FALSE`
  `-MAPPING_CORRECT = "0,1"`

8.7 Adding support for other a.k.a. Custom instruments

It is possible to add support for other instruments[27] without having to edit the distribution files.

To add support for other instruments two files must be made implemented:

- an **OCA file**: this is used by the Data-Organiser to determine which files to process and which files to associate with those files to facilitate the processing
- a **python module**: used by the workflow to adapt as necessary the input data into molecfit compatible input files (and back again if desired) and set parameters as needed for the specific instrument (and mode and/or setting)

These two files should be located following directory of the user running the recipe:

```
${HOME}/KeplerData/workflows/MyWorkflows/my_molecfit
```

Before starting to work on adding support for a not-yet-supported ESO instrument, please contact ESO User Support by creating a ticket at *http://support.eso.org* to check if support for that instrument is not already being developed but just not yet ready for inclusion in the release version.

If you need help implementing support for a Custom instrument, please create a ticket at *http://support.eso.org*.

And if you implement support for an ESO instrument and would like to share it with the community contact ESO User Support by creating a ticket at *http://support.eso.org*.

8.7.1 Custom instrument OCA rule files

The simplist approach to creating an OCA file for another instrument is to start from a template. We recommend starting from the `ANY_wkf.oca` file that comes with the molecfit package. This file is located in the following sub-directory of the molecfit base install[28] directory:

```
share/esopipes/molecfit-<version>/reflex
```

Copy the file to the above directory:

```
cp <base_install_directory>/share/esopipes/molecfit-*/reflex/ANY_wkf.oca \
    ${HOME}/KeplerData/workflows/MyWorkflows/my_molecfit/<INST>_wkf.oca
```

where `<INST>` is the name of the instrument and should be equal to the value of the FITS header keyword `INSTRUME` in the Primary Header of the data files.

[27] Even non-ESO instruments.

[28] For a MacPorts installation the base-install directory is `/opt/local`, for an RPM installation the base-install directory is `/usr` and for an `install_esoreflex` script installation is the `install` subdirectory of the directory where the install script was run.

The `ANY_wkf.oca` already has rules for handling many instruments (besides the ones already supported by the workflow). So you may not need to make any changes at all.

But if you do, bear in mind that the objective of the OCA rules is to identify, using FITS header keyword values (in the Primary headers of the files) those files that can be processed by the workflow. While the majority of the rules already in `ANY_wkf.oca` rely on the keyword "PRO.CATG", any keyword can be used.

Use the examples in `ANY_wkf.oca` and the other `*.oca` files in the same directory to construct a rule(s) that will allow the Data Organizer to identify the relevant files.

The `ANY_wkf.oca` file does nt support the concept of using (say) a telluric star observation to model and calculate the atmosphere and then apply this to the science observations. See the GIRRAFE and/or XSHOOTER OCA files to see how to implement this.

8.7.2 Custom instrument python modules

An instrument specific python module is required to support each instrument.

As for the OCA file, the simplist approach to creating a python module file for another instrument is to start from a template. We recommend starting from the `any.py` file that comes with the molecfit package. This file is located in the same sub-directory of the molecfit base install directory i.e.:

```
share/esopipes/molecfit-<version>/reflex
```

Copy the file to the above directory:

```
cp <base_install_directory>/share/esopipes/molecfit-*/reflex/any.py \
    ${HOME}/KeplerData/workflows/MyWorkflows/my_molecfit/<inst>.py
```

where `<inst>` is the name of the instrument in strict lowercase and should be equal to the value of the FITS header keyword `INSTRUME` in the Primary Header of the data files (in lowercase).

The `any.py` file includes template versions of all the methods that can be used to implement support for an instrument, that will effectively do nothing until modified.

Each method has a short description of its purpose. See the python module files for the supported instruments for examples of how to implement functional the methods.

The GIRAFFE python module provides an example of extracting a 1-D spectrum from a multipe fibre 2-D image FITS file and then applying the results of the molecfit processing back onto each fibre of the original file.

9 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 ([4]). 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 ([4]) 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 ([4]) 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`

10 Troubleshooting

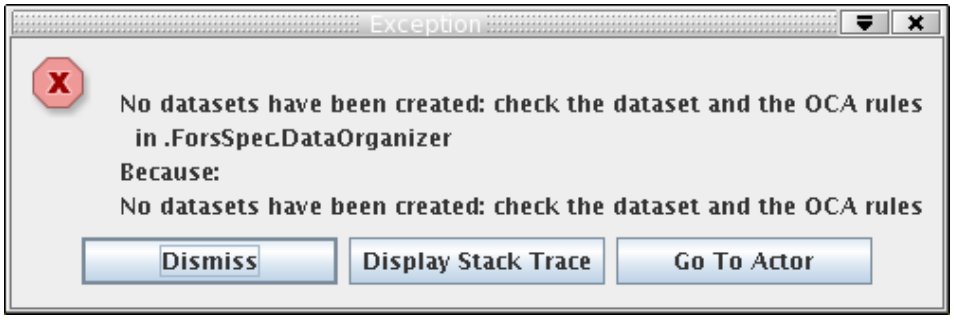


Figure 10.1: TheDataOrganizer interactive window reports an error ":No DataSets have been created, check the data set and the OCA rules."

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 10.1.)**

   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.

4. **How can I correct data of a 1D spectrum for an instrument that is not supported?**

   The molecfit workflow is, in principle, capable of correcting the telluric absorption of any spectrum, provided that it is in a format compatible with the molecfit recipes. We strongly recommend to store your spectrum into a binary table, with all the relevant header information in the primary header, and data in the first extensions.

   For non-ESO instruments (or for ESO instruments that are not recognized by the general OCA rule), please set the keyword `INSTRUME` to `ANY` in the primary header. See Section 8.6 for further details.

A Auto-Molecule

One of the most difficult aspects of implementing support for a new instrument, particularly one with more than one wavelength setup, is configuring the molecules that need to be fitted within the wavelength range of the input data, as well as the wavlength ranges over which to fit the model.

An algorithm to autmomatic set the molecules to be fitted and wavelength ranges over which to make the fits has thus been implemented. Hereafter (and before) this algorithm is referred to as the *auto-molecule algorithm*, or simply `auto-molecule`.

The automatic setting of the fitting wavelength ranges is not expected to be optimal in the majority of cases. The algorithm generally sets a small number (one or two) of rather large intervals. It is then generally advisable to restrict the fitting intervals as few 'short' intervals as possible whilst insuring there is at least one interval containing lines for each molecule being fitted. Ideally the intervals should contain several relatively strong telluric absorption lines and no absorption lines from the source object.

Atlas spectra of each molecule species can be overplot on the `molecfit_model` GUI display to guide the user in setting wavelength ranges containing lines for each molecule to be fit.

The auto-molecule algorithm relies on an atlas constructed for each molecule species supported by molecfit. The atlas is constructed adopting a `REL_COL` value for each molecule of 1.0. The atlas is included in the molecfit datastatic calibration package (`PRO.CATG` = `MOLECULE_ATLAS`).

This atlas is then analysed to determine wavelength regions over which each molecule has absorption lines with strength greater a configurable threshold, typically 2%, producing a static calibration file (`PRO.CATG` = `AUTO_MOLECULES`) that can be used by the molecfit workflow.

As of the current writing the molecfit datastatic calibration package includes an `AUTO_MOLECULES` file for each supported instrument, evaluated over the relevant wavelength regime (UVB, VIS, NIR, VIS+NIR, MIR) for that instrument. All current `AUTO_MOLECULES` files have been calculate for a absorption strength threshold of 2%.

In addition an `AUTO_MOLECULES` file with `INSTRUME` header keyword set to `ANY` has been computed and is included in the molecfit datastatic calibration package for the entire UVB to MIR spectral range (0.3 – 30 μm) with a 2% threshold.

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] Molecfit Pipeline Team, 2020, MOLECFIT Pipeline User Manual, VLT-MAN-ESO-19550-5772, Issue 4.0

[3] Molecfit Pipeline Team, 2022, MOLECFIT Pipeline User Manual, VLT-MAN-ESO-19550-5772, Issue 4.3
28, 29, 32

[4] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, *http://www.eso.org/reflex/*, 2012. VLT-MAN-ESO-19000-5037. 20, 37, 77, 78