EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

Reflex GRAVITY Tutorial and Cookbook

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

Issue 1.8.0

Date 2024-12-03

Prepared: C. Hummel 2024-12-03

Name Date Signature

Approved: –

Name Date Signature

Released: –

Name Date Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
0.1	21/06/2016	All	Initial Release
0.9.2	08/08/2016	All	Updated Release for pipeline version 0.9.2
0.9.3	29/09/2016	All	Updated Release for pipeline version 0.9.3
1.0.0	17/10/2016	All	Updated Release for pipeline version 1.0.0
1.0.5	26/05/2017	All	Change to Cookbook for pipeline version 1.0.5
1.0.11	14/11/2017	All	Change to Cookbook for pipeline version 1.0.11
1.2.5	01/05/2019	All	Change to Cookbook for pipeline version 1.2.5

This page was intentionally left blank

Contents

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 reduction of GRAVITY single and dual mode observations only via the GRAVITY Reflex workflow. For more detail on the pipeline, the user is referred to the pipeline manual and the GRAVITY user manual ([2][2]) and to the ESO instrument web pages [3] for more information on the instrument itself as well as a summary of available documentation, recent news, and tools. The cookbook aspects derive from further guidance on GRAVITY data reduction and analysis in general.

The quick start section (see Section 5) describes the minimum effort to get started, and it makes up only two pages of text in this tutorial. User support for this software is available at `https://support.eso.org`

[1] *https://www.eso.org/sci/archive/calselectorInfo.html*
[2] available at: https://www.eso.org/sci/facilities/paranal/instruments/gravity/doc
[3] https://www.eso.org/sci/facilities/paranal/instruments/gravity.html

2 Workflow Status

The GRAVITY Reflex workflow, in its current version, is capable, together with its underlying GRAVITY pipeline, of delivering calibrated data products. The Reflex workflows are built upon the GRAVITY pipeline delivered by LESIA, Observatoire de Paris.

The first step of the GRAVITY Reflex workflow is to organise the data of this instrument into an associated, organised, and classified structure including for each science or interferometric calibrator file the required instrument calibration files with matching spectral resolution and integration time. The user will be warned if any calibration frames are missing.

The GRAVITY Reflex workflow, *gravity_wkf*, will correct the frames for their dark level and structure, flat-field the data, and compute a wavelength solution, A large number of data products are created and retained for the user to assess the quality of the pipeline processing.

An alternative workflow, *gravity_mastercalib*, allows to reduce data with the P2VM master calibration file which can be downloaded from the archive. This allows the user to skip the resource-intensive P2VM computation step (you would need at least 8GB of RAM in high resolution mode).

Calibrating the science observations is done with a separate workflow, *gravity_viscal*.

During the processing within the Reflex workflow, the user has the ability to modify a number of pipeline parameters in order to optimise the data processing.

During the pipeline development and the experience of GRAVITY use, the pipeline parameters have been set to default values that deliver the best results for the most cases. However, the user should make an effort to adjust and experiment with the parameters to optimize the results.

3 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 pipelines and Reflex can be installed from the ESO `macports` repositories that support macOS platforms, the and the `rpm/yum` repositories that support Fedora and CentOS platforms. 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.

3.1 Installing `Esoreflex` workflows via `macports`

This method is supported for the macOS operating system. It is assumed that macports (*https://www.macports.org*) is installed. Please read the full documentation at *https://www.eso.org/sci/software/pipelines/installation/macports.html*, which also describes the versions of macOS that are currently supported.

3.2 Installing `Esoreflex` workflows via `rpm/yum/dnf`

This method is supported for Fedora and CentOS platforms and requires sudo rights. Please read the full documentation at *https://www.eso.org/sci/software/pipelines/installation/rpm.html*, which also describes the versions of Fedora and CentOS that are currently supported.

3.3 Installing `Esoreflex` 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 2.11.3` may be found at:
https://www.eso.org/sci/software/pipelines/reflex_workflows

To install the `Reflex 2.11.3` software and demo data, please follow these instructions:

1. From any directory, download the installation script:

   ```
   wget https://eso.org/sci/software/pipelines/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.

4 Demo Data

Together with the pipeline you can also choose to receive a demo data set that allows you to run the `Reflex GRAVITY` workflow without any changes in parameters. This way you have data sets to experiment with before you start to work on your own data.

Note that you will need a minimum of ∼5 GB, ∼9 GB and ∼8 GB of free disk space for the directories `<download_dir>`, `<install_dir>` and `<data_dir>`, respectively.

The raw input consists of OBJ-SKY-OBJ-SKY sequences of:

1. two calibrator OBs of HIP 64314, and
2. one science OB of a double star, HD 114529 ($\rho = 143.0$ mas, $\theta = 132.6°$, secondary flux fraction = 24%).

The data set also includes the DARK, P2VM, FLAT, WAVE, and WAVESC files. The raw tutorial data sets are summarized in Table 4.1. The list of files as shown in the table can be obtained by executing the following command (part of the ESO SciSoft collection) in the demo data directory:

```
dfits *.fits | fitsort obs.name dpr.catg dpr.type tpl.expno
ins.filt1.name ins.filt2.name dit
```

FILT1 refers to the Wollaston prism, which is "IN" when recording both polarizations and "OUT" when recording the sum of both signals. FILT2 refers to the spectral resolution. Note the two DARK frames with different DIT, one belonging to the science frames, the other one to the P2VM frames.

Table 4.1: The GRAVITY Reflex workflow tutorial data set, single field mode data with medium spectral resolution and no polarization.

File	OBS.NAME	DPR.CATG	DPR.TYPE	EXPNO	FILT1.NAME	FILT2.NAME	DIT
GRAVI.2016-06-22T16:26:14.536.fits	Calibration	CALIB	DARK	1	OUT	MED	0.3
GRAVI.2016-06-22T16:27:29.540.fits	Calibration	CALIB	FLAT	2	OUT	MED	0.3
GRAVI.2016-06-22T16:28:05.542.fits	Calibration	CALIB	FLAT	3	OUT	MED	0.3
GRAVI.2016-06-22T16:28:41.544.fits	Calibration	CALIB	FLAT	4	OUT	MED	0.3
GRAVI.2016-06-22T16:29:17.546.fits	Calibration	CALIB	FLAT	5	OUT	MED	0.3
GRAVI.2016-06-22T16:30:02.549.fits	Calibration	CALIB	WAVE,SC	6	OUT	MED	0.3
GRAVI.2016-06-22T16:35:38.568.fits	Calibration	CALIB	WAVE	7	OUT	MED	0.3
GRAVI.2016-06-22T16:40:23.583.fits	Calibration	CALIB	P2VM	8	OUT	MED	0.3
GRAVI.2016-06-22T16:42:59.592.fits	Calibration	CALIB	P2VM	9	OUT	MED	0.3
GRAVI.2016-06-22T16:45:35.601.fits	Calibration	CALIB	P2VM	10	OUT	MED	0.3
GRAVI.2016-06-22T16:48:11.610.fits	Calibration	CALIB	P2VM	11	OUT	MED	0.3
GRAVI.2016-06-22T16:50:47.619.fits	Calibration	CALIB	P2VM	12	OUT	MED	0.3
GRAVI.2016-06-22T16:53:23.628.fits	Calibration	CALIB	P2VM	13	OUT	MED	0.3
GRAVI.2016-06-22T17:11:47.690.fits	Calibration	CALIB	DARK	1	OUT	MED	10.0
GRAVI.2016-06-23T00:19:55.884.fits	CAL_HIP64314	CALIB	OBJECT,SINGLE	1	OUT	MED	10.0
GRAVI.2016-06-23T00:24:37.900.fits	CAL_HIP64314	CALIB	SKY,SINGLE	2	OUT	MED	10.0
GRAVI.2016-06-23T00:29:22.916.fits	CAL_HIP64314	CALIB	OBJECT,SINGLE	3	OUT	MED	10.0
GRAVI.2016-06-23T00:34:04.932.fits	CAL_HIP64314	CALIB	SKY,SINGLE	4	OUT	MED	10.0
GRAVI.2016-06-23T00:47:52.978.fits	SCI_HD114529	SCIENCE	OBJECT,SINGLE	1	OUT	MED	10.0
GRAVI.2016-06-23T00:52:34.994.fits	SCI_HD114529	SCIENCE	SKY,SINGLE	2	OUT	MED	10.0
GRAVI.2016-06-23T00:57:20.010.fits	SCI_HD114529	SCIENCE	OBJECT,SINGLE	3	OUT	MED	10.0
GRAVI.2016-06-23T01:02:02.026.fits	SCI_HD114529	SCIENCE	SKY,SINGLE	4	OUT	MED	10.0
GRAVI.2016-06-23T01:17:05.077.fits	CAL_HIP64314	CALIB	OBJECT,SINGLE	1	OUT	MED	10.0
GRAVI.2016-06-23T01:21:47.092.fits	CAL_HIP64314	CALIB	SKY,SINGLE	2	OUT	MED	10.0
GRAVI.2016-06-23T01:26:35.108.fits	CAL_HIP64314	CALIB	OBJECT,SINGLE	3	OUT	MED	10.0
GRAVI.2016-06-23T01:31:20.125.fits	CAL_HIP64314	CALIB	SKY,SINGLE	4	OUT	MED	10.0

5 Quick Start: Reducing The Demo Data

For the user who is keen on starting reductions without being distracted by detailed documentation, we describe the steps to be performed to reduce the science data provided in the GRAVITY demo data set supplied with the esoreflex 2.11.3 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 GRAVITY by typing:

```
esoreflex gravity_wkf&
```

   Alternatively, you can type only the command esoreflex the empty canvas will appear (Figure 5.1) 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 GRAVITY workflow is shown in Figure 8.1.

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 8.4) that lists the datasets along with the values of a selection of useful header keywords[4]. 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 7.2.2.

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 8.7 shows the Product Explorer window. A full description of the *Product Explorer* will be presented in Section 7.2.3.

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 GRAVITY workflow that merit a look at the rest of this tutorial.

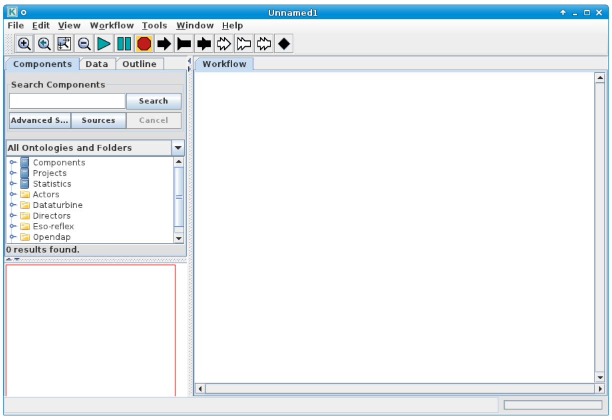


Figure 5.1: *The empty* Reflex *canvas.*

[4]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.

6 About the main esoreflex canvas

6.1 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.

6.2 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.

6.3 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.

7 The GRAVITY Workflow

The GRAVITY 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.

7.1 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 7.2.5), 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.

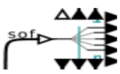
7.2 Workflow Actors

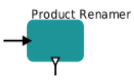
7.2.1 Simple Actors

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

- 

 - The `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).

7.2.2 DataSetChooser

The `DataSetChooser` displays the DataSets available in the "Select Data Sets" window, activating vertical and horizontal scroll bars if necessary (Fig. 8.4).

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[5], 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.

7.2.3 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. 8.7).

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:

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

- 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. 8.7). 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).

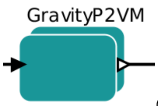
2. To inspect the desired file, proceed as follows:

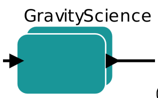
- Navigate through the data reduction cascade in the ProductExplorer by clicking on the files.
- Select the file to be inspected and click with the mouse right-hand button. The available options are:
  - Options available always:
    - Copy full path. It copies the full name of the file onto the clipboard. 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.

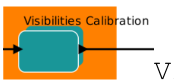
7.2.4 GRAVITY-specific actors: the workflow data-reduction cascade

The present GRAVITY 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:

- 

`GravityDark`: it executes the recipe *gravity_dark* . It processes the dark frames to create a master dark needed both for the recipe *gravity_p2vm* and for *gravity_vis*. The master dark frames need to have the same DIT as the corresponding science/calibration frames.
- 

`GravityP2VM`: it executes the recipe *gravity_p2vm*. It requires products of *gravity_dark* as input, as well as raw P2VM, FLAT, and WAVE files.
- 

`GravityScience`: it executes the recipe *gravity_vis*. It requires products of *gravity_dark* as input, as well as reduced BAD, P2VM, FLAT, and WAVE files.
- 

`Visibilities Calibration`: it executes the recipe *gravity_viscal*. It requires products of *gravity_vis* as input, for a science target and one or more calibrator observations.

  This is an interactive actor, meaning that an apposite interacting window will appear allowing the used to inspected the products and, eventually, to re-run the recipe with modified parameters.

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

As noted in the Quick Start Section 5, the workflow will then proceed through its remaining processing steps and write out all pipeline products to the end products directory (specified by the parameter END_PRODUCTS_DIR under "Setup Directories" in the workflow canvas). The science data products from the tutorial data set are summarized in section 9 in table 9.2. The intermediate pipeline calibration products can be found in subdirectories of the TMP_PRODUCT_DIR and are summarized in section 9 and in table

7.2.5 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.*

8 Reducing and Calibrating Your Own Science Data with Reflex

8.1 Available Reflex workflows

There are two workflows available for the reduction and one workflow for the calibration of the visibility data. While the workflow *gravity_wkf* shown in Fig. 8.1 performs all reduction steps using the raw data, the workflow *gravity_mastercalib* shown in Fig. 8.2 uses the P2VM master calibration files downloaded from the ESO archive (asscociated *processed* calibrations), allowing you to bypass the resource-heavy step of the P2VM computation. The workflow *gravity_viscal* in Fig. 8.3 calibrates the reduced science files.

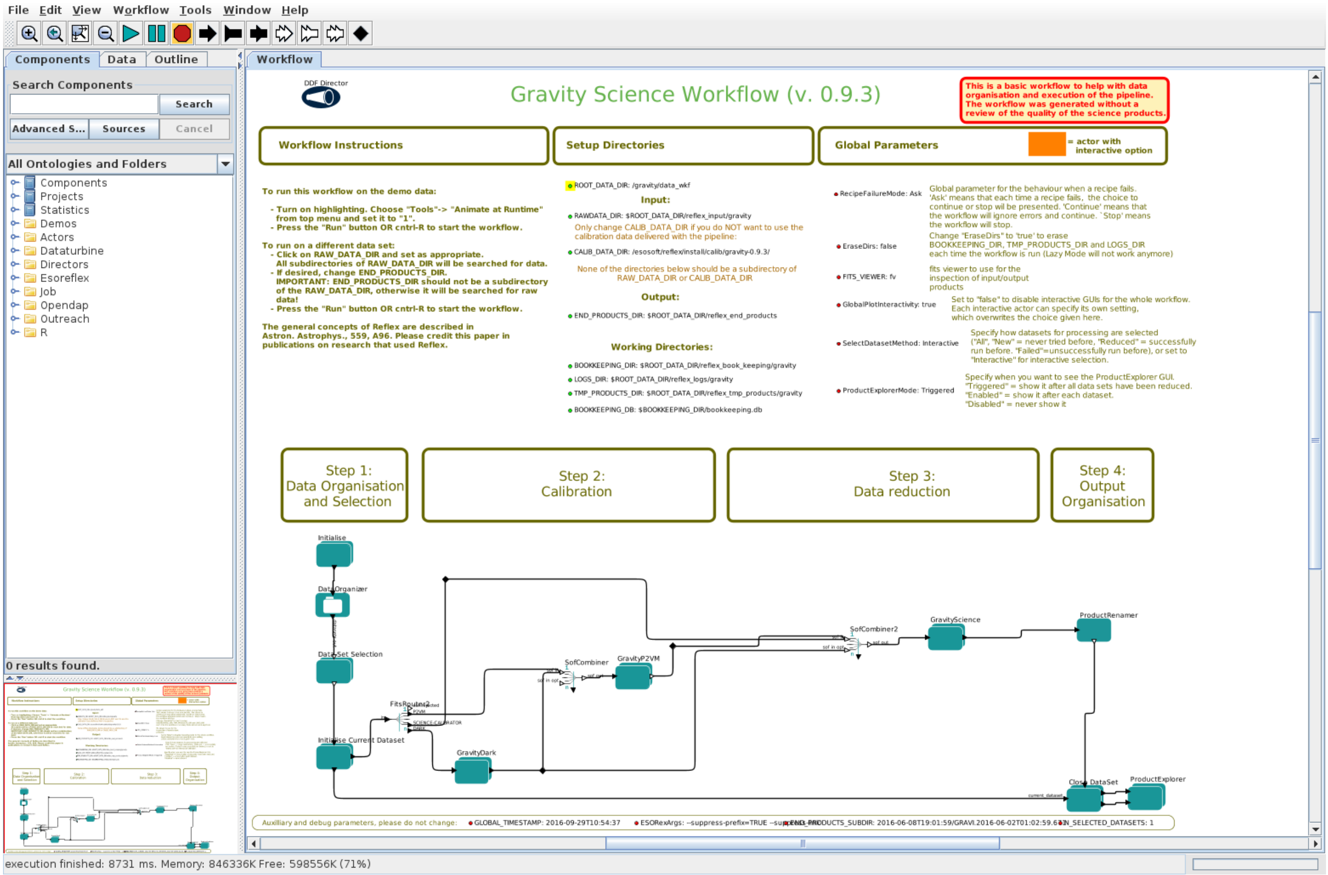


Figure 8.1: *gravity_wkf workflow layout.*

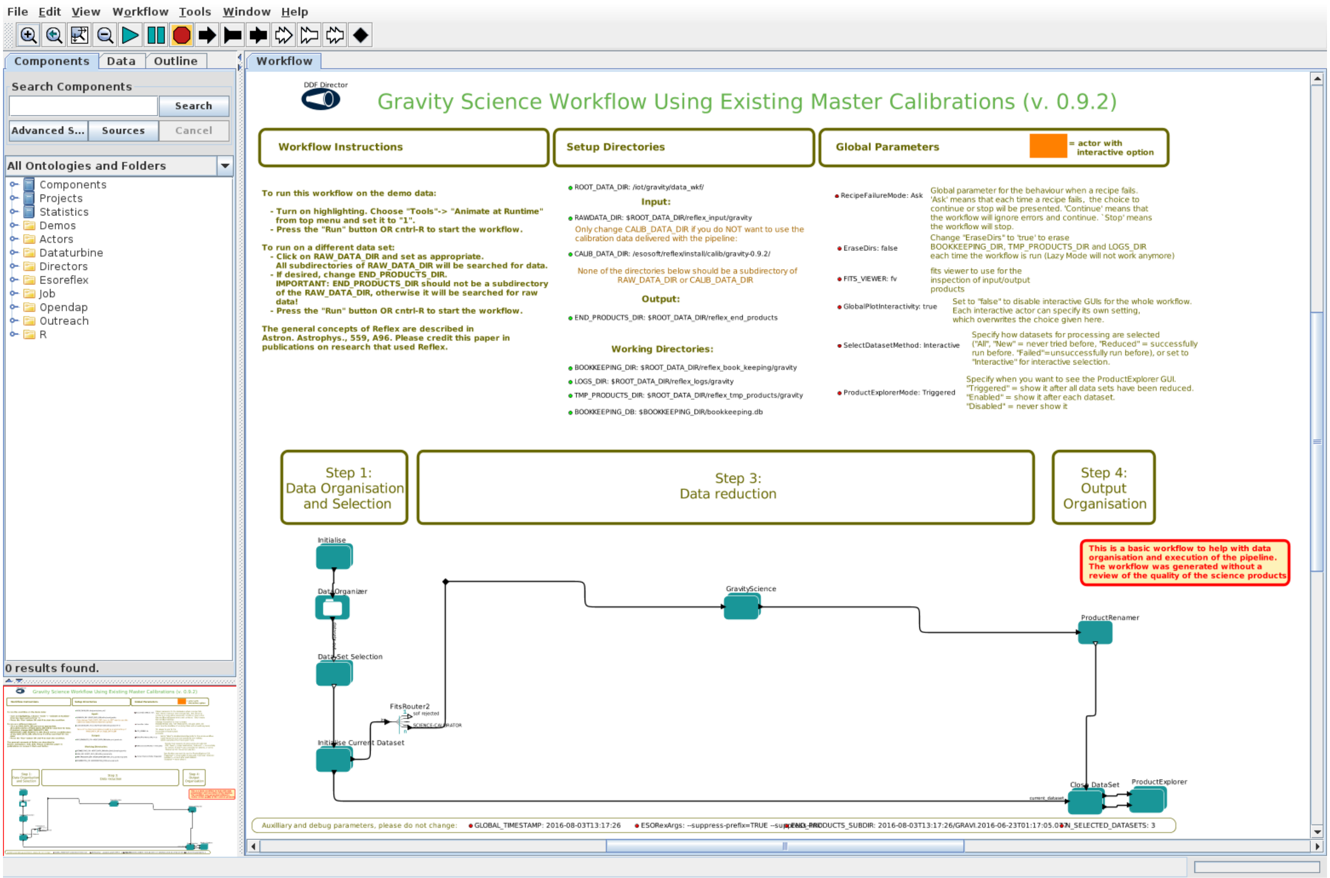


Figure 8.2: *gravity_mastercalib workflow layout.*

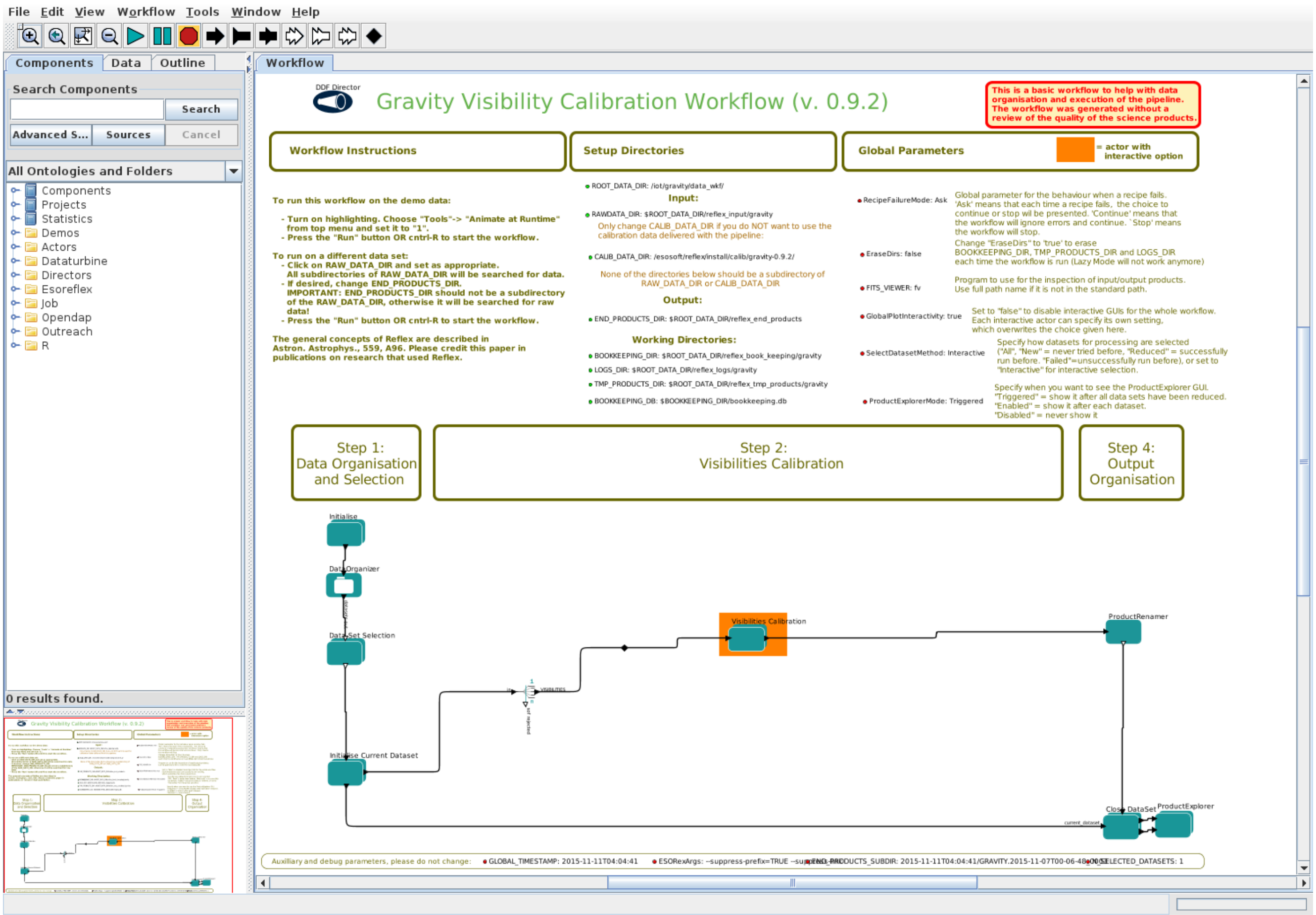


Figure 8.3: *gravity_viscal workflow layout.*

8.2 Specifying data directories and selecting files

To reduce your own science data, simply change the paths to the root (optional) and data directories. Under the root directory, Reflex will create sub-directories which will contain temporary and end products, as well as book keeping and log files. The data directory, normally also under the root directory, contains directories with your raw files downloaded from the ESO archive. The paths are defined at the top of the workflow window in the area labeled `Setup Directories`. Simply double click on the `RAWDATA_DIR`, enter the path to your raw science directory and then start the workflow in the same way as you did for the tutorial demo data. In case the data sets listed in the first window created by the work flow (Fig. 8.4) are greyed out, calibration files are missing (hovering with the mouse over the grey file entry will give more details). You can click the entry and a GUI opens up showing the dependency tree of the science (or calibrator) file on calibrations (Fig. 8.4). Please note that the pipeline step which computes the P2VM calibration file requires about 8 GB of memory for the demo data (medium resolution, no polarization), about 12 GB for high resolution (no polarization), and up to about 20 GB for high resolution with polarization.

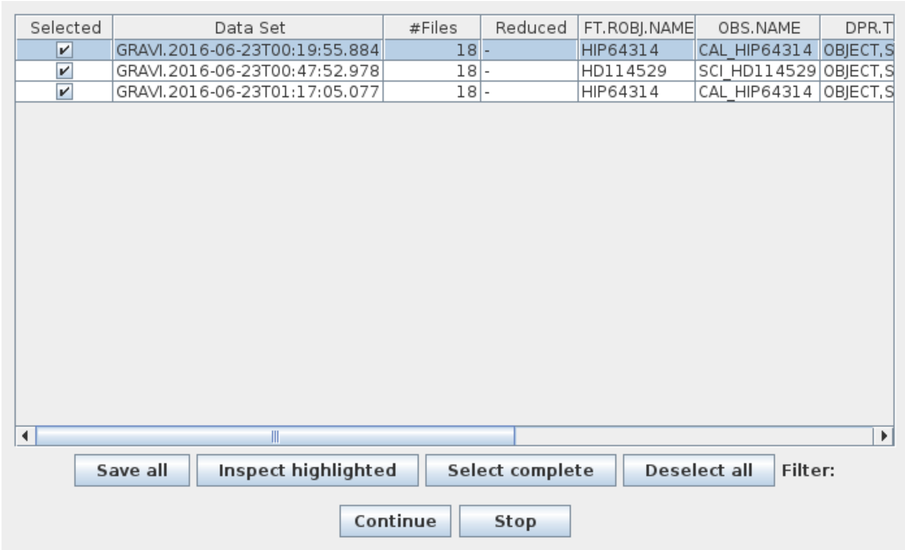


Figure 8.4: The "Select Datasets" pop-up window.

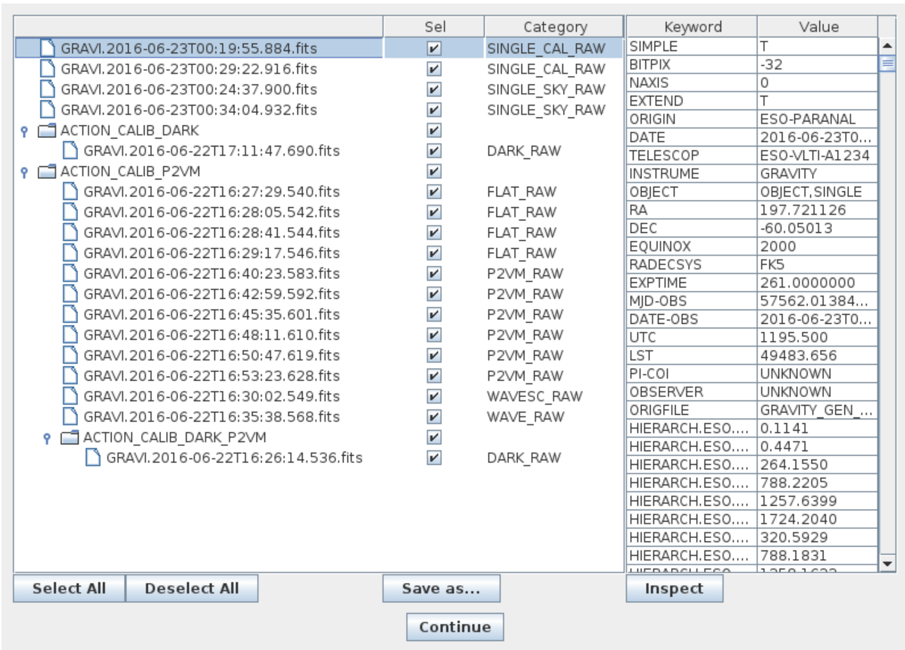


Figure 8.5: The "Select Frame" pop-up window, obtained after pressing the "Inspect highlighted" button in the "Select Datasets" window.

8.3 Correction of systematic errors in the visibilities

The visibility amplitudes are affected by bias and coherence time. The former is related to photon noise and needs to be subtracted from the (square) modulus of the complex visibility, the latter can be corrected by multiplying the visibility with a factor larger than 1, called vFactor, which is derived from the GRAVITY fringe tracker phase RMS.

In version 1.8.0 of the pipeline, if the vFactor correction results in visibilities significantly larger than unity, the correction must be switched off. To do this, right-click with the mouse on the GravityScience actor and select "Open actor". The workflow canvas of this actor will be displayed, and here you right-click on the gravity_vis actor to select "Configure actor". This will display a window like the one shown in Fig. 8.6. To disable the vFactor correction, change the value of vis-correction-sc (recipe_param_26) from VFACTOR to NONE. Please note that turning the vFactor correction off is not recommended in general.

Likewise, if you needed to disable the bias correction, you would set the value of parameter debias-sc (recipe_param_23) to "false". However, please note that the bias correction is a fundamental and well established correction which should only be disabled if you have very good evidence for it not to work in your specific case.

.

8.4 Removing instrumental signatures from the source spectrum

GRAVITY records the source spectrum in addition to the source fringes. The former is affected by the wavelength-dependent throughput, and to correct for this, the value of the parameter flat-flux (recipe_param_29) must be set to "true".

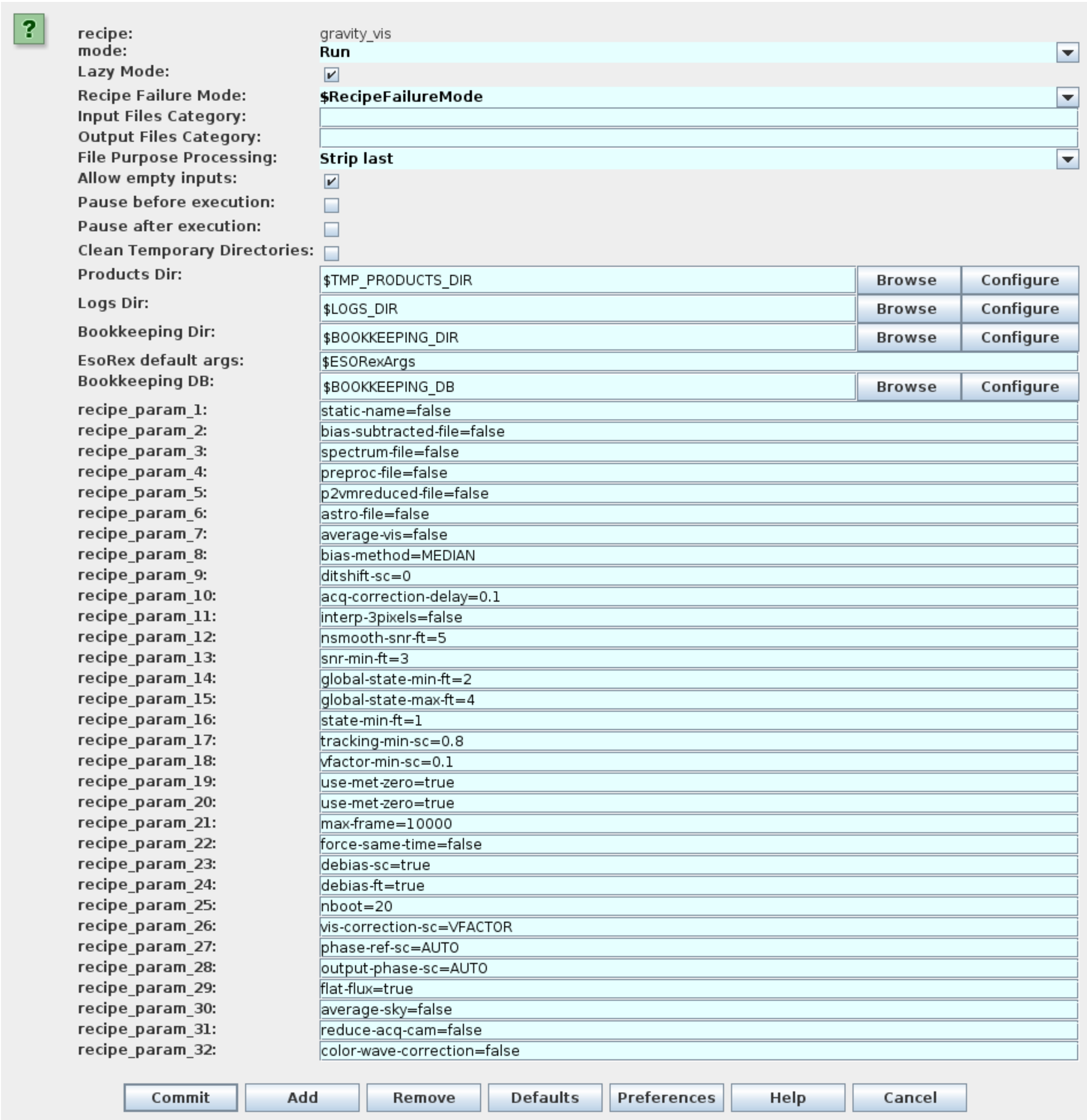


Figure 8.6: *The interactive window to edit parameters of recipe gravity_vis.*

8.5 Examining the workflow results

When the workflow has finished, the Product Explorer window opens (Fig. 8.7). Select a data file and unfold the file tree in the "Provenance Tree" window. This provides information on the dependency of product files on the calibration files and other files from recipes executed before. You can inspect a data file by clicking the "Inspect with..." button, and entering the path to your favourite FITS file viewer (e.g., *fv*). Examples of the master flat frame (Fig. 8.8) and a master dark frame (Fig. 8.9) are shown below.

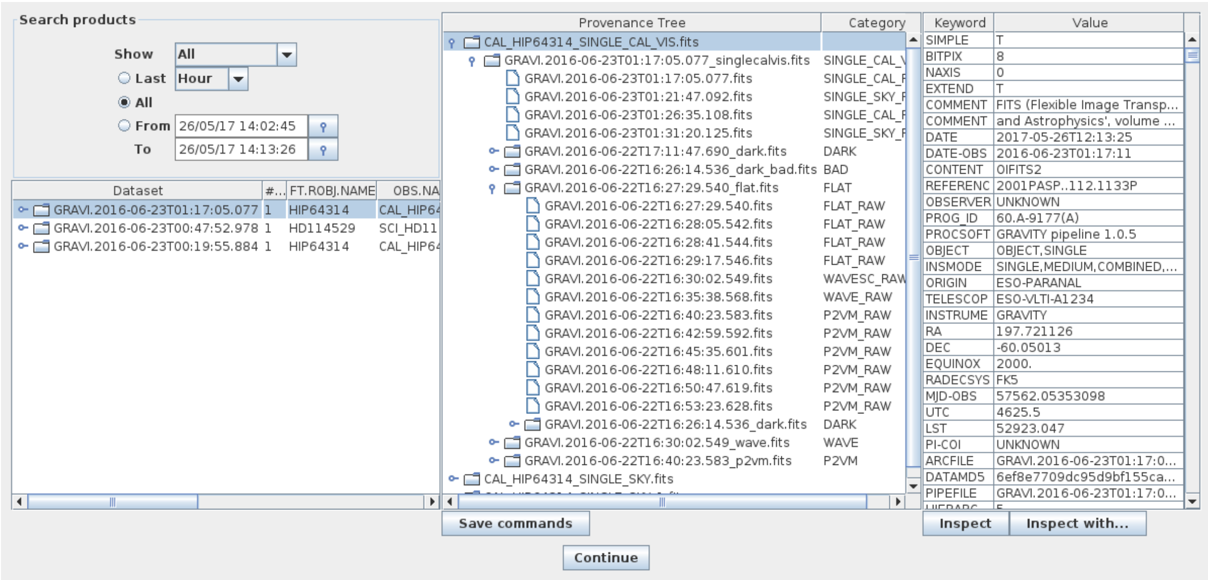


Figure 8.7: *The GRAVITY product explorer for the gravity_vis workflow.*

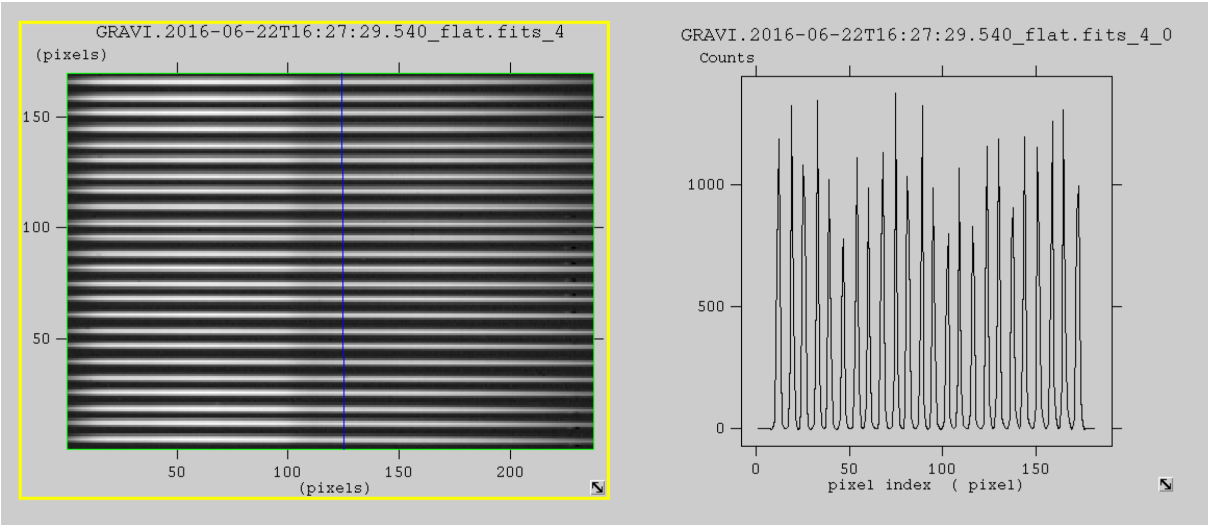


Figure 8.8: *GRAVITY master flat image, stored in the IMAGING_DATA_SC binary extension. The horizontal axis is the wavelength axis, the 24 output channels are arranged along the vertical axis.*

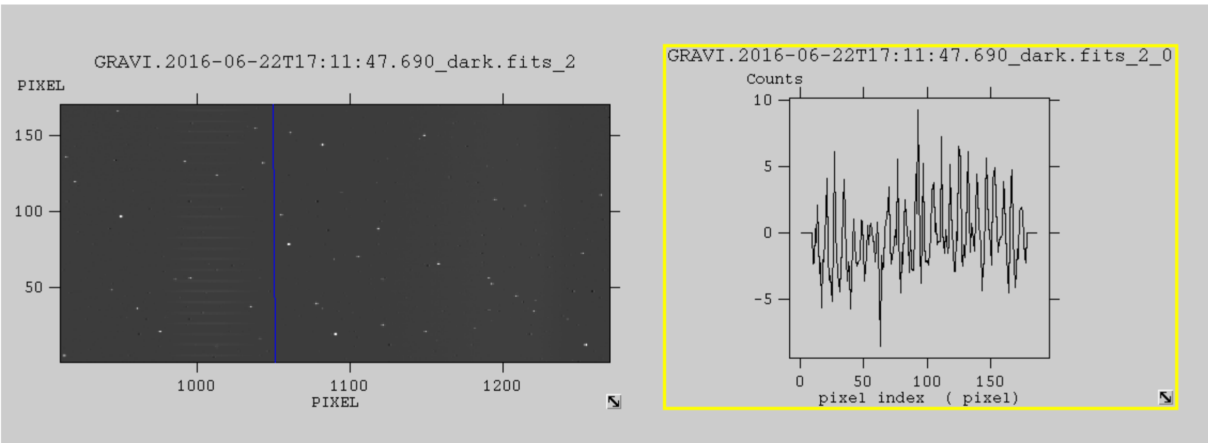


Figure 8.9: *GRAVITY master dark image, stored in the IMAGING_DATA_SC binary extension. The horizontal axis is the wavelength axis, the 24 output channels are arranged along the vertical axis.*

8.6 Calibrating your visibility data

The workflow *gravity_viscal* allows you to select (aside from the science file) one or more calibrator files which will define the transfer function (TF, i.e., the visibility measured on a calibrator, corrected for the effect of a non-zero diameter of the calibrator). The demo data set includes two calibrators, one measured before and one after the science target. Thus, the TF will be linearly interpolated to the epoch of the science observations. When running this recipe, you will be presented with a plot window showing the TF for each of the six baselines of GRAVITY, as shown in Fig. 8.10. The calibrator diameter information is taken from *GRAVI_FAINT_CALIBRATORS.fits* in the Reflex directory *install/calib/gravity-1.1.2/cal*. Please note that the calibrated data files are not in the *reflex_end_products* directory, but under the *reflex_tmp_products/gravity_viscal/gravity_v* directory, in a folder named with the date and time when it was created.

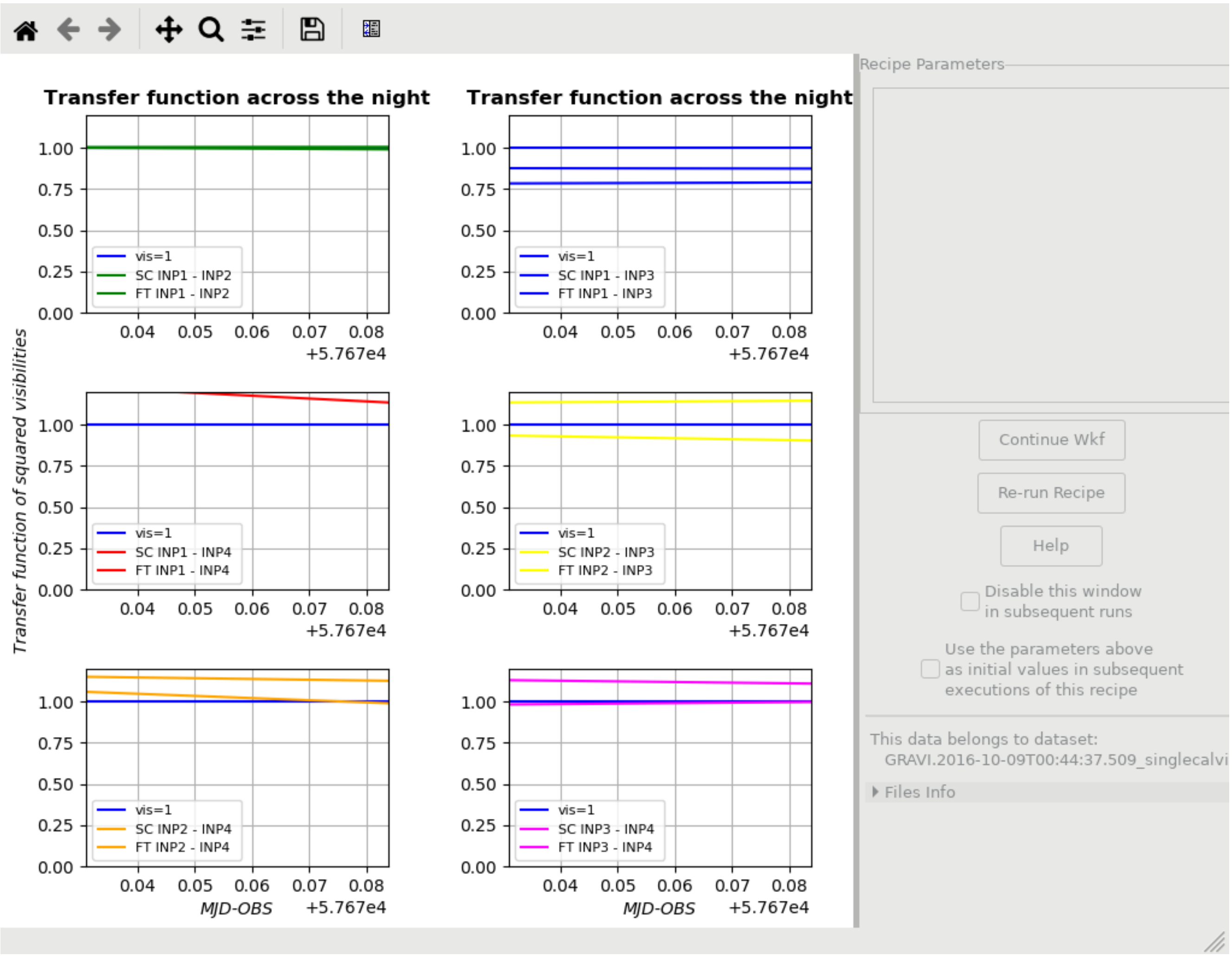


Figure 8.10: *gravity_viscal workflow plot of the FT and SC channel transfer functions.*

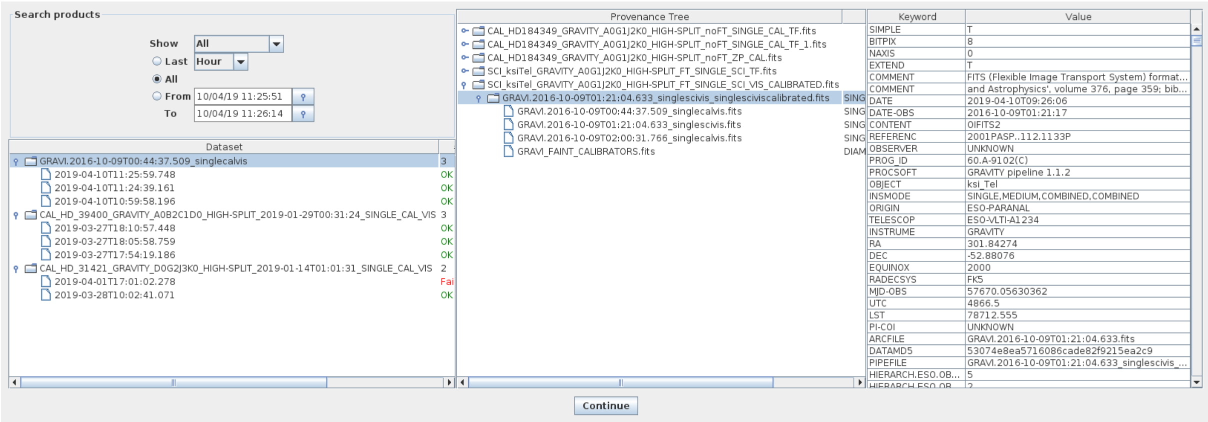


Figure 8.11: *The GRAVITY product explorer for the gravity_viscal workflow.*

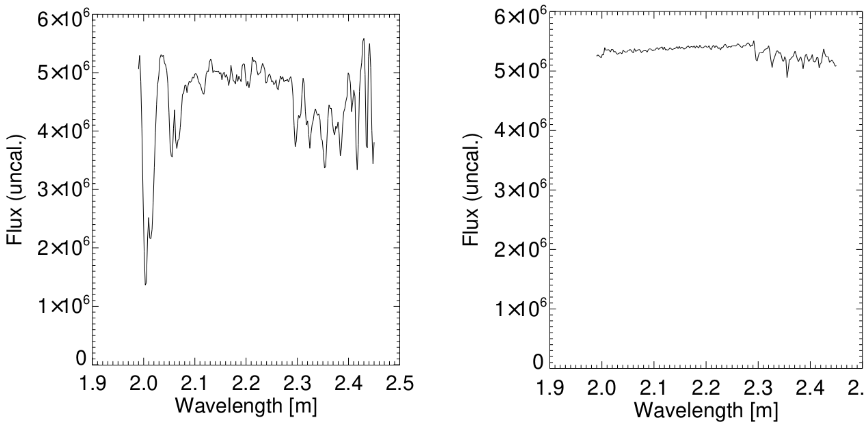


Figure 8.12: Uncalibrated (left) and calibrated (right) medium resolution spectra of the science target, produced by gravity_viscal. The deep absorption lines in the uncalibrated spectrum are telluric features of CO_2 and H_2O.

8.7 Calibrating your flux spectra

The workflow *gravity_viscal* also allows you to divide the science spectrum by the calibrator spectrum. If your calibrator had a featureless spectrum, you would get the best correction of the telluric features possible. However, this is almost never the case since nearly featureless spectra are only displayed by early-type stars, and these are usually poor calibrators due to the high fraction of multiples in these classes. To have the workflow do the division, you should set the value of the parameter calib-flux to *true* (recipe_parameter_7).

8.8 Telluric corrections of the flux spectra with Molecfit

Molecfit [6] is a tool to fit abundances of atmospheric molecules to their telluric absorption lines in astronomical spectra. These can then be used in turn to compute the telluric absorption over the entire wavelength range covered by the spectrum for the purpose of removing the telluric absorption lines. In the following, we show an example computed for the high resolution mode of GRAVITY. In this example, the data file (*gravity.dat*) is a text file with two colums, wavelength (in microns) and flux. In the K band we only fit the carbondioxide, water, and methane lines. A Gaussian line profile is adopted, whose width changes with wavelength. Atmospheric parameters such as temperature, pressure, humidity, can be extracted from the FITS headers of the GRAVITY data files. An example parameter file is given below. The only parameter values which need to be updated are `cont_const`, the approximate level of the continuum flux, observation date and time, telescope altitude (i.e., pointing), and the weather parameters.

```
filename: gravity.dat
trans: 1
columns: Wavelength Flux NULL NULL
default_error: 1.0
wlgtomicron: 1.0
vac_air: air
wrange_include: gravity_include.dat
wrange_exclude: gravity_exclude.dat
ftol: 0.01
xtol: 0.01
list_molec: H2O CO2 CH4
fit_molec: 1 1 1
relcol: 1.0 1.06 1.0
flux_unit: 0
fit_back: 0
telback: 0.1
fit_cont: 1
cont_n: 1
cont_const: 110000.0
fit_wlc: 1
wlc_n: 1
wlc_const: 0.0
fit_res_box: 0
relres_box: 0.0
fit_res_gauss: 1
res_gauss: 1.4
fit_res_lorentz: 0
res_lorentz: 0.5
kernmode: 0
kernfac: 30.0
varkern: 1
```

[6] https://www.eso.org/sci/software/pipelines/skytools/molecfit

```
obsdate:  57849.000
obsdate_key: MJD-OBS
utc:  5826.8243
utc_key: UTC
telalt:  68.923724
telalt_key: ESO TEL ALT
rhum: 32.0000
rhum_key: ESO TEL AMBI RHUM
pres: 744.380
pres_key: ESO TEL AMBI PRES START
temp: 14.4900
temp_key: ESO TEL AMBI TEMP
m1temp: 14.4900
m1temp_key: ESO TEL TH M1 TEMP
geoelev: 2635.0
geoelev_key: ESO TEL GEOELEV
longitude: -70.4032
longitude_key: ESO TEL GEOLON
latitude: -24.6258
latitude_key: ESO TEL GEOLAT
slitw: 0.4
slitw_key: ESO INS SLIT1 WID
pixsc: 0.086
pixsc_key: NONE
ref_atm: equ.atm
gdas_dir: data/profiles/grib
gdas_prof: auto
layers: 1
emix: 5.0
pwv: -1.
end
```

The file *gravity_include.dat* is shown below. It contains the wavelength intervals including the most prominent CO2, CH4, and H2O lines for the fit of the abundances of these species.

```
# CO2
1.99 2.08
# CH4
2.315 2.321
2.363 2.380
#
# H2O
2.41 2.45
```

The file *gravity_exclude.dat* is shown below. It contains the wavelength intervals including the most common intrinsic lines of the late-type calibrators or science targets which must not be fit.

```
# He line
2.058 2.060
# Br-g line
2.150 2.190
# CO band heads
2.292 2.298
2.321 2.327
2.350 2.356
2.381 2.387
```

The result of an application of Molecfit is shown in Fig. 8.13. Even though the most prominent telluric features have been removed from the spectra, the result shows that instrumental residuals from the flat-fielding remain, which are only removed if one divides the science spectrum by the one of the calibrator.

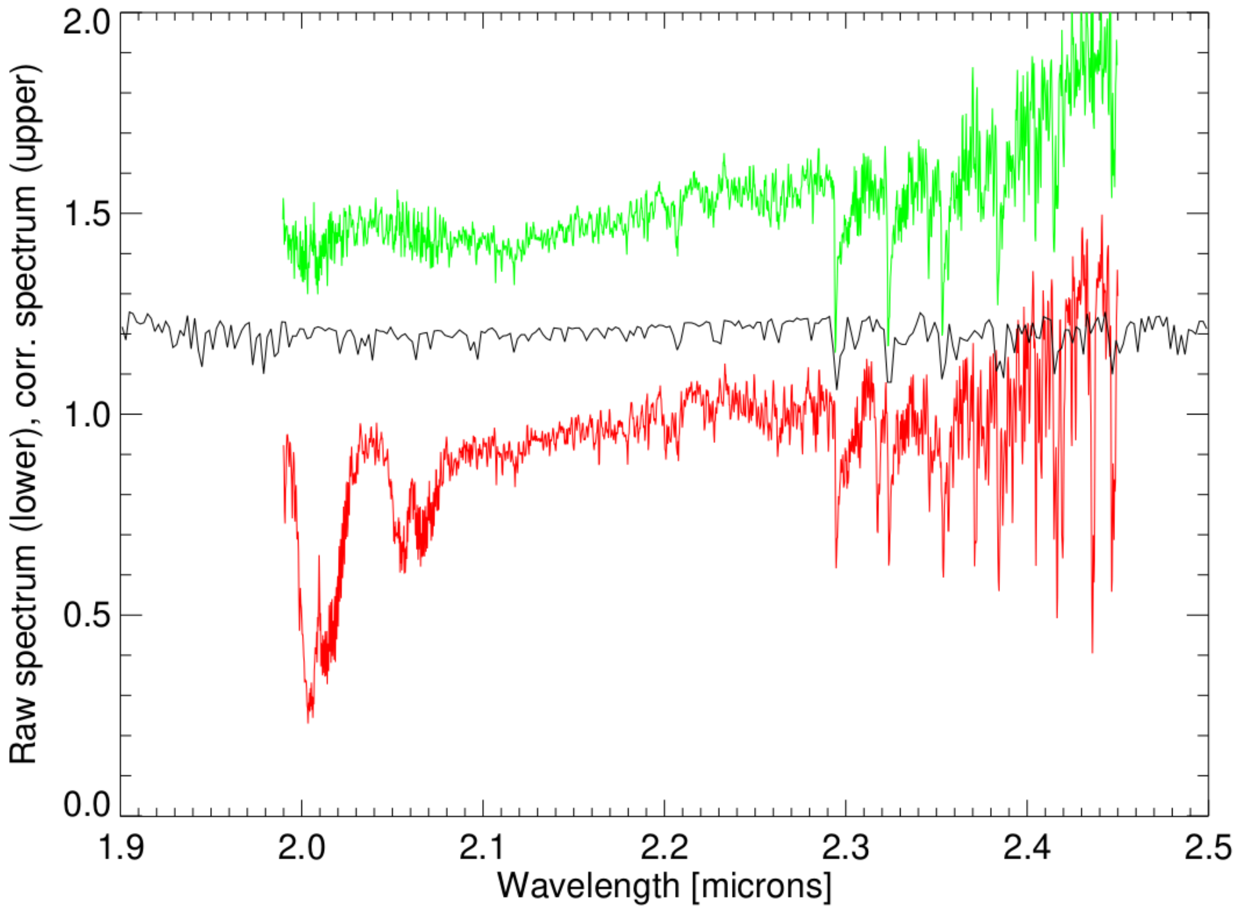


Figure 8.13: Telluric correction computed with Molecfit. The lower (red) spectrum is of the raw flux, the upper (green) spectrum is the raw flux divided by the atmospheric transmission fit to the telluric lines. The middle line is a (lower resolution) spectrum from the NextGen library [3] for the spectral type of the observed star.

9 GRAVITY Reduced Data Description

A number of intermediate pipeline products from the tutorial data set can be found in subdirectories of the TMP_PRODUCT_DIR. These master calibration files are summarized in Table 9.1.

Table 9.1: The GRAVITY Reflex workflow tutorial data set: calibration products

File	PRO.CATG	Description
gravity_p2vm:		
GRAVI.2016-06-22T16:26:14.536_dark_bad.fits	BAD	bad pixel map
GRAVI.2016-06-22T16:27:29.540_flat.fits	FLAT	flat field
GRAVI.2016-06-22T16:35:38.568_wave.fits	WAVE	wavelength calibration
GRAVI.2016-06-22T16:40:23.583_p2vm.fits	P2VM	p2vm
gravity_dark:		
GRAVI.2016-06-22T16:26:14.536_dark.fits	DARK	dark frame

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

The science data products from the two tutorial data sets are summarized in tables 9.2. Their description will be done in Section 9.1.

Table 9.2: The GRAVITY Reflex workflow tutorial data set: uncalibrated visibility products

File	PRO.CATG	Description
GRAVI.2016-06-22T23:07:43.639/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-22T23:07:43.639/CAL_HIP64314_SINGLE_CAL_VIS.fits	SINGLE_CAL_VIS	uncalibrated averaged visibility data
GRAVI.2016-06-22T23:35:46.735/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-22T23:35:46.735/CAL_HIP64314_SINGLE_CAL_VIS.fits	SINGLE_CAL_VIS	uncalibrated averaged visibility data
GRAVI.2016-06-22T23:52:43.791/SCI_HD114529_SKY.fits	SKY	sky frame
GRAVI.2016-06-22T23:52:43.791/SCI_HD114529_SINGLE_SCI_VIS.fits	SINGLE_SCI_VIS	uncalibrated averaged visibility data
GRAVI.2016-06-23T00:19:55.884/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-23T00:19:55.884/CAL_HIP64314_SINGLE_CAL_VIS.fits	SINGLE_CAL_VIS	uncalibrated averaged visibility data
GRAVI.2016-06-23T00:47:52.978/SCI_HD114529_SKY.fits	SKY	sky frame
GRAVI.2016-06-23T00:47:52.978/SCI_HD114529_SINGLE_SCI_VIS.fits	SINGLE_SCI_VIS	uncalibrated averaged visibility data
GRAVI.2016-06-23T01:17:05.077/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-23T01:17:05.077/CAL_HIP64314_SINGLE_CAL_VIS.fits	SINGLE_CAL_VIS	uncalibrated averaged visibility data

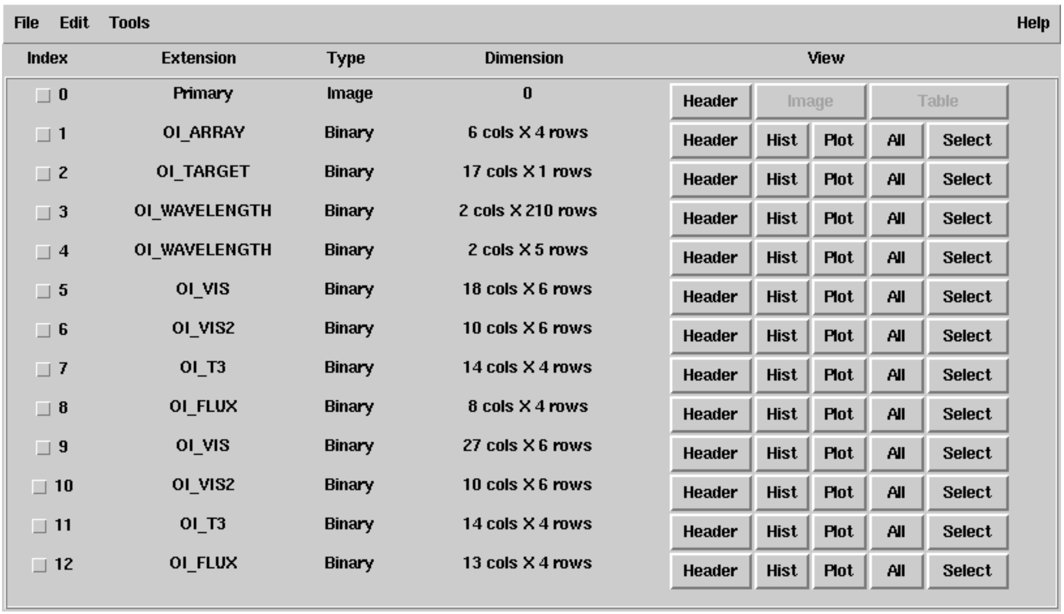
Table 9.3: The GRAVITY Reflex workflow tutorial data set: calibrated visibility of the science target

File	PRO.CATG	Description
gravity_viscal:		
CAL_HIP64314_SINGLE_CAL_TF_1.fits	SINGLE_CAL_TF	Transfer function of calibrator
CAL_HIP64314_SINGLE_CAL_TF.fits	SINGLE_CAL_TF	Transfer function of calibrator
CAL_HIP64314_ZP_CAL.fits	ZP_CAL	Zero-point calibration (photometry)
SCI_HD114529_SINGLE_SCI_TF.fits	SINGLE_SCI_TF	Transfer function interpolated for science target
SCI_HD114529_SINGLE_VIS_CALIBRATED.fits	SINGLE_SCI_VIS_CALIBRATED	calibrated science visibility

9.1 Description of final products

In this section we provide a very brief description of the workflow science products that are stored in the `reflex_end_products` directory. They are produced by the pipeline recipes *gravity_vis*. For further information, please consult the GRAVITY pipeline manual. In the list, the products are identified by their PRO.CATG keyword.

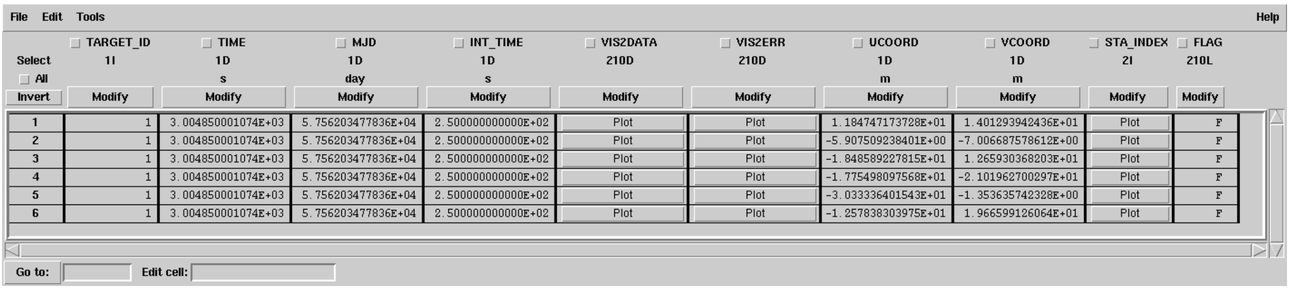
- `SINGLE_CAL_VIS`. Product of the recipe *gravity_vis* for a CAL observation. This file should conform to the OIFITS2 file format used for interferometric data [1]. If the visibilities of the calibrator are corrected for the non-zero diameter of the calibrator star, the resulting visibility amplitudes are called the *transfer function* (TF).
- `SINGLE_SCI_VIS`. Product of the recipe *gravity_vis* for a SCI observation. This file should conform to the OIFITS2 file format used for interferometric data [1]. The visibility data are averaged, but not calibrated.
- `SINGLE_SCI_VIS_CALIBRATED`. Product of the recipe *gravity_viscal* for a SCI observation. This file should conform to the OIFITS2 file format used for interferometric data [1]. The visibility data are averaged and were calibrated by dividing the science target visibilities by the calibrator transfer function interpolated for the epoch of the science observation. Fig. 9.1 shows the structure of an OIFITS2 file and Figs. 9.2 and 9.3 show the extenstion table for the (squared) visibility amplitudes and a plot for one of the baselines.



File Edit Tools Help

Index	Extension	Type	Dimension	View				
0	Primary	Image	0	Header	Image	Table		
1	OI_ARRAY	Binary	6 cols X 4 rows	Header	Hist	Plot	All	Select
2	OI_TARGET	Binary	17 cols X 1 rows	Header	Hist	Plot	All	Select
3	OI_WAVELENGTH	Binary	2 cols X 210 rows	Header	Hist	Plot	All	Select
4	OI_WAVELENGTH	Binary	2 cols X 5 rows	Header	Hist	Plot	All	Select
5	OI_VIS	Binary	18 cols X 6 rows	Header	Hist	Plot	All	Select
6	OI_VIS2	Binary	10 cols X 6 rows	Header	Hist	Plot	All	Select
7	OI_T3	Binary	14 cols X 4 rows	Header	Hist	Plot	All	Select
8	OI_FLUX	Binary	8 cols X 4 rows	Header	Hist	Plot	All	Select
9	OI_VIS	Binary	27 cols X 6 rows	Header	Hist	Plot	All	Select
10	OI_VIS2	Binary	10 cols X 6 rows	Header	Hist	Plot	All	Select
11	OI_T3	Binary	14 cols X 4 rows	Header	Hist	Plot	All	Select
12	OI_FLUX	Binary	13 cols X 4 rows	Header	Hist	Plot	All	Select

Figure 9.1: *Main structure of OIFITS2 file with binary extension tables.*



File Edit Tools Help

Select	TARGET_ID	TIME	MJD	INT_TIME	VIS2DATA	VIS2ERR	UCOORD	VCOORD	STA_INDEX	FLAG
	1I	1D	1D	1D	210D	210D	1D	1D	2I	210L
All		s	day	s			m	m		
Invert	Modify	Modify	Modify	Modify	Modify	Modify	Modify	Modify	Modify	Modify
1	1	3.004850001074E+03	5.756203477836E+04	2.500000000000E+02	Plot	Plot	1.184747173728E+01	1.401293942436E+01	Plot	F
2	1	3.004850001074E+03	5.756203477836E+04	2.500000000000E+02	Plot	Plot	-5.907509238401E+00	-7.006687578612E+00	Plot	F
3	1	3.004850001074E+03	5.756203477836E+04	2.500000000000E+02	Plot	Plot	-1.848589227815E+01	1.265930368203E+01	Plot	F
4	1	3.004850001074E+03	5.756203477836E+04	2.500000000000E+02	Plot	Plot	-1.775498097568E+01	-2.101962700297E+01	Plot	F
5	1	3.004850001074E+03	5.756203477836E+04	2.500000000000E+02	Plot	Plot	-3.033336401543E+01	-1.353635742328E+00	Plot	F
6	1	3.004850001074E+03	5.756203477836E+04	2.500000000000E+02	Plot	Plot	-1.257838303975E+01	1.966599126064E+01	Plot	F

Go to: Edit cell:

Figure 9.2: *OIVIS2 table.*

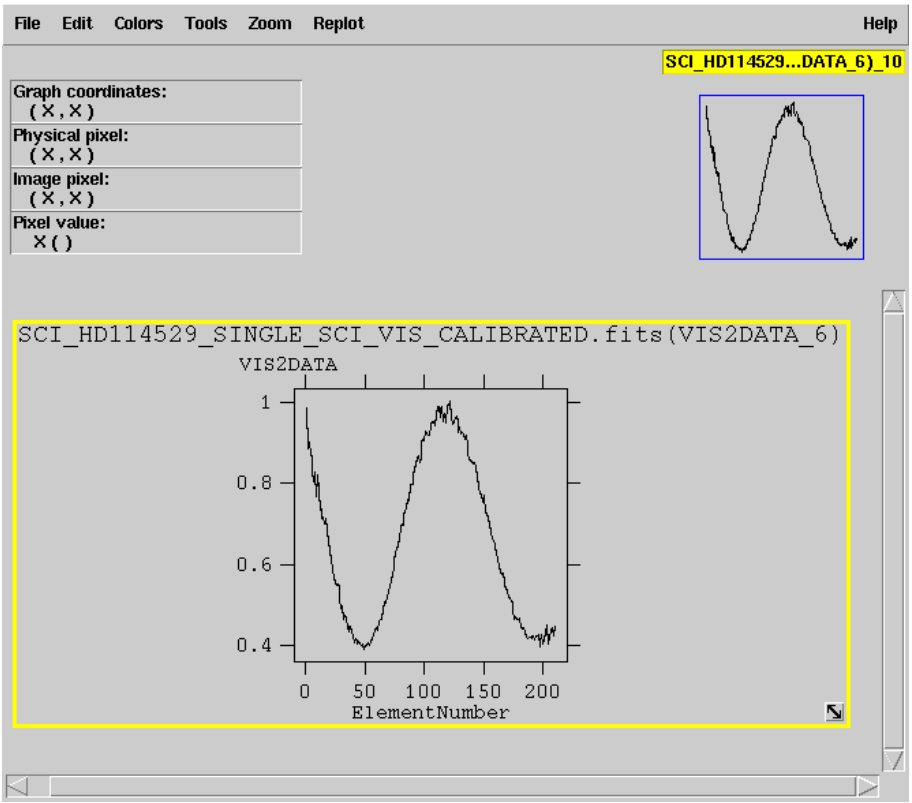


Figure 9.3: *Plot of the squared visiblity amplitude for baseline 6.*

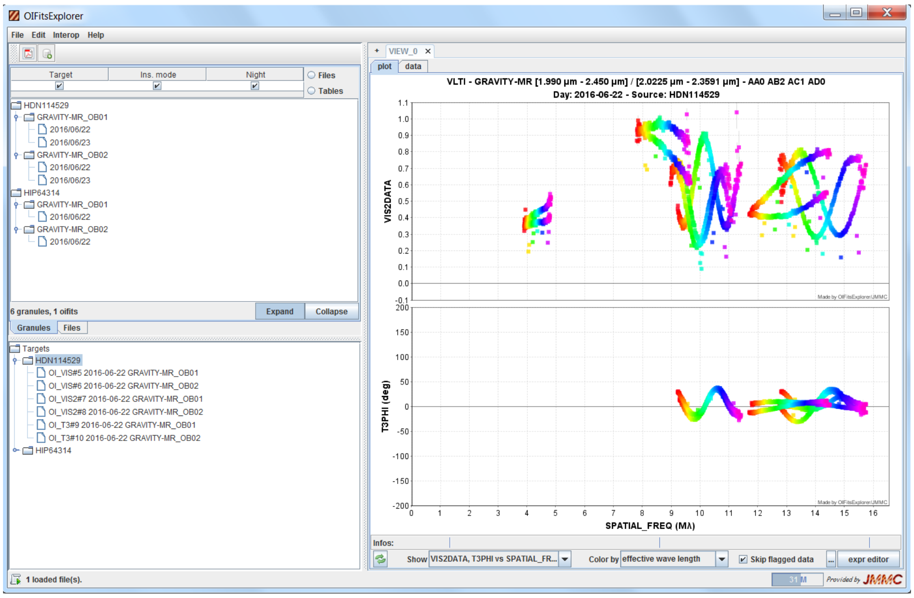


Figure 9.4: OIFitsExplorer view of the demo data (all science files combined).

One can use the OIFitsExplorer (Fig. 9.4) (`http://www.jmmc.fr/oifitsexplorer_page.htm`) to examine the data and LitPro (`http://www.jmmc.fr/litpro_page.htm`) to fit simple models.

9.2 Calibration of flux spectra

The column named "FLUX" in the "OI_FLUX" table (of the science detector) contains the reduced fluxes from each of the four telescopes for every integration.

10 Reducing Your Own Science Data with Python tools

The GRAVITY consortium developed a set of Python scripts which interface with pipeline recipes to reduce and analyze data. These tools are not maintained by ESO and no support is provided by ESO. The scripts can be downloaded from *http://version-lesia.obspm.fr/repos/DRS_gravity/python_tools/* or with the following command:

```
svn co https://version-lesia.obspm.fr:/repos/DRS_gravity/python_tools
```

The scripts will be located in a local directory name *python_tools*. Please consult the README file in that directory for instructions on how to change the PATH environment variable so that the scripts can be found by python.

To reduce all demo data at once, change the working directory to the *data_wkf/reflex_input/gravity/gravity-demo-reflex-0.5* created when you downloaded the data. Since the science and calibration data files are located in verious sub-directories, create symbolic links to them in the working with the following comand (include the "dot" at the end!):

```
ln -s */*.fits .
```

Then run the following command to reduce all files and store the results in a local directory named *reduced*:

```
python run_gravi_reduce.py
```

The following command, if run in the reduced files directory, produces very informative PDF files on the (uncalibrated) results. The same command, when run in *reflex_tmp_products/gravity/gravity_p2vm_1* (then choosing the sub-directory with the date/time stamp corresponding to the desired execution of the work flow), produces a PDF file with various plots useful for quality control of the P2VM (see the first page in Fig. 10.1).

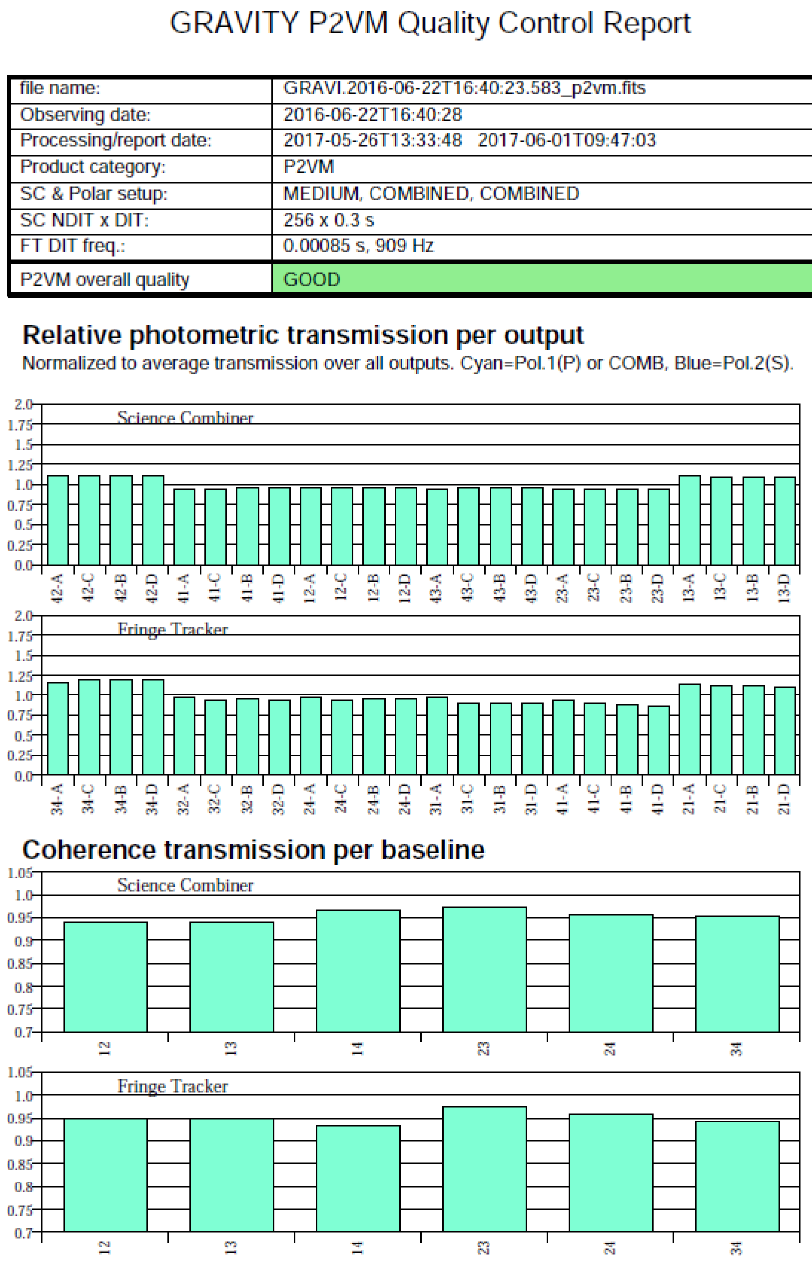
```
python run_gravi_visual.py
```

A useful script is also *quicklook.py* (*http://github.com/amerand/GRAVIQL*) which plots visibilities, phases, and fluxes for reeduced data files.

Finally, the following command calibrates the visibilities of the science targets:

```
python run_gravi_trend.py
```

It creates two directories, *trends* and *calibrated*. Files *GRAVI.????-??-??T??:??:??.???_*_singlesciviscalibrated.fits* contain the calibrated visibilities of the science targets. Alternatively, the reduced data can be calibrated using the Reflex it gravity_viscal workflow.



GRAVITY P2VM Quality Control Report

file name:	GRAVI.2016-06-22T16:40:23.583_p2vm.fits
Observing date:	2016-06-22T16:40:28
Processing/report date:	2017-05-26T13:33:48 2017-06-01T09:47:03
Product category:	P2VM
SC & Polar setup:	MEDIUM, COMBINED, COMBINED
SC NDIT x DIT:	256 x 0.3 s
FT DIT freq.:	0.00085 s, 909 Hz
P2VM overall quality	GOOD

Figure 10.1: *The first page of the P2VM report produced by run_gravi_visual.py. Note the overall quality rating box near the top of the page.*

11 Astrometry Cookbook

11.1 Introduction

GRAVITY is capable of observing two targets simultaneously if they are close together on the sky (a few seconds of arc). The templates enabling this mode have names starting with "GRAVITY_dual". Consequently, they allow to specify two sets of coordinates labeled SC and FT, for science and fringe tracker, respectively, where the coordinates of the latter are encoded in offsets of Right Ascension and Declination. Fringes of both targets are recorded, before swapping SC and FT and recording another batch of fringes. During the swap, the metrology beams of GRAVITY which monitor internal path length differences must not be interrupted. The swap allows to determine the so-called metrology zeropoint by computing the difference of the SC-FT delay before and after the swap.

In this cookbook we outline the data reduction steps and the underlying methodology for the derivation of the precise relative positions of SC and FT targets. The cookbook is based on the Python tools written by M. Nowak for the analysis of GRAVITY astrometry data. These tools are neither maintained nor guaranteed by ESO and no support is provided by ESO. Following the tradition of cookbooks as well, we encourage the reader here and there to visualize the data from the pipeline products as well and study their interdependences and relationships.

11.2 Reducing raw astrometric data with the ESO pipeline

Fortunately, the details of reducing the data and removing all the instrumental signatures are of no concern to us, as these have been taken care of by the pipeline developers. All what is needed is to specify in the parameter set of the *gravity_vis* recipe `astro-file=true` and `reduce-acq-cam=true` before running the workflow. It does happen occasionally that reduction of the acquisition images fails, in which case the resulting image scale parameters in the output file `HIERARCH.ESO.QC.ACQ.FIELDi.SCALE` and `HIERARCH.ESO.QC.ACQ.FIELDi NORTH ANGLE` have to be computed manually. You may go to support.eso.org for help in this case.

The products created by this recipe can be found as usual under the *reflex_end_products* directory, which are files of type `ASTROREDUCED`.

11.3 Outline of the analysis steps

Fundamentally, knowing the orientation of the baseline relative to the target and the path length difference (delay measured in meters) between the two arms of the interferometer while tracking the fringes, will allow to obtain the projection of the position of the target orthogonal to the (known) baseline orientation. By combining several measurements on other baselines with different orientations simultaneously, the absolute positions can be derived. In the case of GRAVITY, which measures differential positions of the fringe patterns of the two targets, a relative position (separation and position angle) is derived.

11.3.1 The narrow-angle baseline

The almost-parallel beams of the two targets propagated by the VLTI to the GRAVITY instrument are separated by a roof-top beamsplitter, and then fed to two beamcombiners, one for the FT target and the other one for the

SC target. By separating the beams, a very short baseline, the so-called narrow angle (NA) baseline has been realized which makes GRAVITY in effect an interferometr in itself. The VLTI infrastructure of telescopes, beam transport, and delay lines are merely used to propagate two targets (instead of the usually single target) and take out most of the optical path length difference between the VLTI telescopes (corresponding to the wide-angle baselines).

The relative position of the two targets is then constrained by the delay measured for the narrow-angle baseline of GRAVITY by short fiber-delay lines. A laser-based metrology systems controls the geometry of the GRAVITY beamcombiner internally and in relation for the telescope structure.

11.4 Python tools

Tools written in Python 3 by M. Nowak for the analysis of GRAVITY astrometry data are available and are installed as follows:

```
git clone https://gitlab.obspm.fr/mnowak/cleanGravity
git clone https://gitlab.obspm.fr/mnowak/exogravity.git
cd exogravity
git checkout ESOswap
```

Additional Python libraries might have to be installed as well, if not already available:

pip install –user ruamel.yaml
pip install –user notebook

The notebook can be used to view tutorials, as follows (other examples are available as well):

cd exogravity/tutorial
jupyter-notebook swap_example.ipynb

11.5 Data files for a swap observation

The list of raw files included in the astrometry demo data set is summarized in Table 11.1. The list was produced with the following command (part of the ESO SciSoft collection):

```
dfits *.fits | fitsort ins.sobj.name ft.robj.name ins.sobj.x ins.sobj.y ins.sobj.s
```

These data need to be reduced with the pipeline, as described above, before using the Python tools for the astrometic analysis.

11.5.1 Analysis

As a first step, a configuration file needs to be created with information on the type of observation (SWAP or other), the directory with the pipeline products (files of type ASTROREDUCED, based on the OIFITS format standard), and other miscellaneous parameters. It is assumed in the following that the directory *astroreduced*

Table 11.1: The GRAVITY Reflex workflow astrometry tutorial data set, off-axis dual field mode data with medium spectral resolution and split polarization. The units of the X and Y offsets are in milli-arcseconds.

File	INS.SOBJ.NAME	FT.ROBJ.NAME	SOBJ.X	SOBJ.Y	SOBJ.SWAP
GRAVI.2018-11-22T03:09:53.338.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T03:13:05.347.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T03:16:20.355.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T03:23:53.374.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T03:27:02.382.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T03:33:41.399.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T03:36:50.407.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T03:45:20.428.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T03:48:29.437.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T03:55:05.453.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T03:58:14.462.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T04:08:11.487.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T04:11:23.495.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T04:17:50.511.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T04:20:59.520.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T04:27:44.537.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T04:30:53.545.fits	GJ65A	GJ65B	207.19	-2166.8	YES
GRAVI.2018-11-22T04:38:32.564.fits	GJ65B	GJ65A	-207.19	2166.8	NO
GRAVI.2018-11-22T04:41:41.572.fits	GJ65B	GJ65A	-207.19	2166.8	NO

with the pipeline products is a sub-direcotry of the working directory. Furthermore, it is assumed that the directory *exogravity* is above the working directory.

```
python ../exogravity/create_config.py datadir="./astroreduced" output="gj65.yml"
swap_target=GJ65 suffix="ASTROREDUCED" calib_strategy="none" extension=11
```

For files with split polarimetric output, the OI_VIS extension can be either 11 or 12, otherwise it would be 10. Please ignore the fact that the extension number is not identical with the extension number listed in the OIFITS file. The messages issued by the script should state that files of type SWAP have been found, which is a standard observation sequence for dual field observations of two targets (FT and SC) which are both bright enough for fringe tracking.

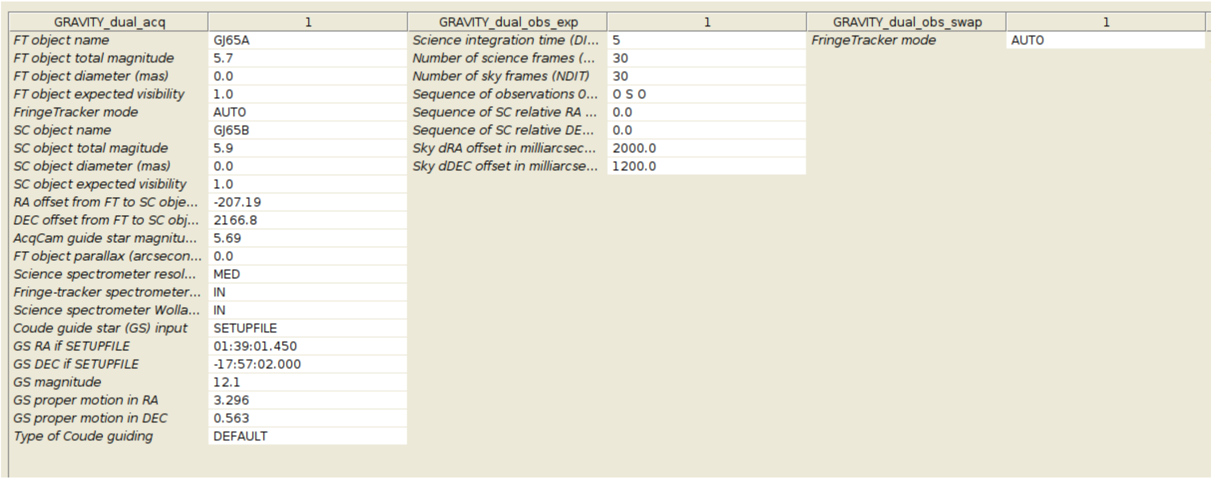
The messages from the script should be like this:

```
t=1.69s Loading ./astroreduced/GJ65_2018-11-22T03:09:44_ASTROREDUCED.fits
t=2.36s Target is GJ65; Fiber distance is 2176.68 mas.  Target is GJ65.  This
is a SWAP!
```

Please note that we defined the swap target as GJ65, and not GJ65A or GJ65B. The reason is that the script takes the target info from the FITS header of the OIFITS file, and *not* from the OI_TARGET extension. The script first calculates the distance between the SC and FT targets from the keywords INS.SOBJ.X/Y and assumes the observation is of the star if the distance is less than 10 mas. Otherwise, it is assumed the observation is of a "planet", unless the `swap_target` parameter is defined and equal to the object name defined in the file header. The following Figure 11.1 shows the OB definition for the observations analayzed here. Note that GS RA and DEC values must be identical to the coordinates specified in the target tab the OB.

Once the configuration file has been created, the following python script is executed, after creating a directory named here as `figures`:

```
python ../exogravity/swap_reduce.py config_file="gj65.yml"
```



GRAVITY_dual_acq	1
FT object name	GJ65A
FT object total magnitude	5.7
FT object diameter (mas)	0.0
FT object expected visibility	1.0
FringeTracker mode	AUTO
SC object name	GJ65B
SC object total magnitude	5.9
SC object diameter (mas)	0.0
SC object expected visibility	1.0
RA offset from FT to SC obje...	-207.19
DEC offset from FT to SC obj...	2166.8
AcqCam guide star magnitu...	5.69
FT object parallax (arcsecon...	0.0
Science spectrometer resol...	MED
Fringe-tracker spectrometer...	IN
Science spectrometer Wolla...	IN
Coude guide star (GS) input	SETUPFILE
GS RA if SETUPFILE	01:39:01.450
GS DEC if SETUPFILE	-17:57:02.000
GS magnitude	12.1
GS proper motion in RA	3.296
GS proper motion in DEC	0.563
Type of Coude guiding	DEFAULT

GRAVITY_dual_obs_exp	1
Science integration time (DI...	5
Number of science frames (...	30
Number of sky frames (NDIT)	30
Sequence of observations O...	O S O
Sequence of SC relative RA ...	0.0
Sequence of SC relative DE...	0.0
Sky dRA offset in milliarcsec...	2000.0
Sky dDEC offset in milliarcse...	1200.0

GRAVITY_dual_obs_swap	1
FringeTracker mode	AUTO

Figure 11.1: *OB definition of a dual field observation including a swap template..*

```
raguess=-208 decguess=2167 ralim=\[-218,-198\] declim=\[2157,2177\]
figdir=figures -gofast
```

If successful, the last messages by the script should look like this:

```
RA grid:  [-218.00, -198.00] with 100 points
DEC grid:  [2157.00, 2177.00] with 100 points
Astrometric solution found using NO SWAP file:
RA=-209.474747 mas, DEC=2166.696970 mas
Astrometric solution found using SWAP files:
RA=207.898990 mas, DEC=-2166.949495 mas
```

The script will deposit image files in the `figures` folder, showing the achieved fits to the complex coherent flux data (Fig. 11.2) and the astrometric χ^2 maps (Fig. 11.3 of each observation.

A swap observation as shown above can be used also as a calibration of the metrology zeropoint for another dual field observation of a target with a companion which is too faint to track. Such an observation should be concatenated with the science target. In this case, after having run the `swap_reduce.py` script on the data of the calibration binary, the following scripts are used to first create the phase reference file and then use it to compute the astrometric solution for the science observation:

```
python ../exogravity/create_phase_reference.py config_file=gj5.yml
python ../exogravity/astrometry_reduce.py config_file=gj65.yml
```

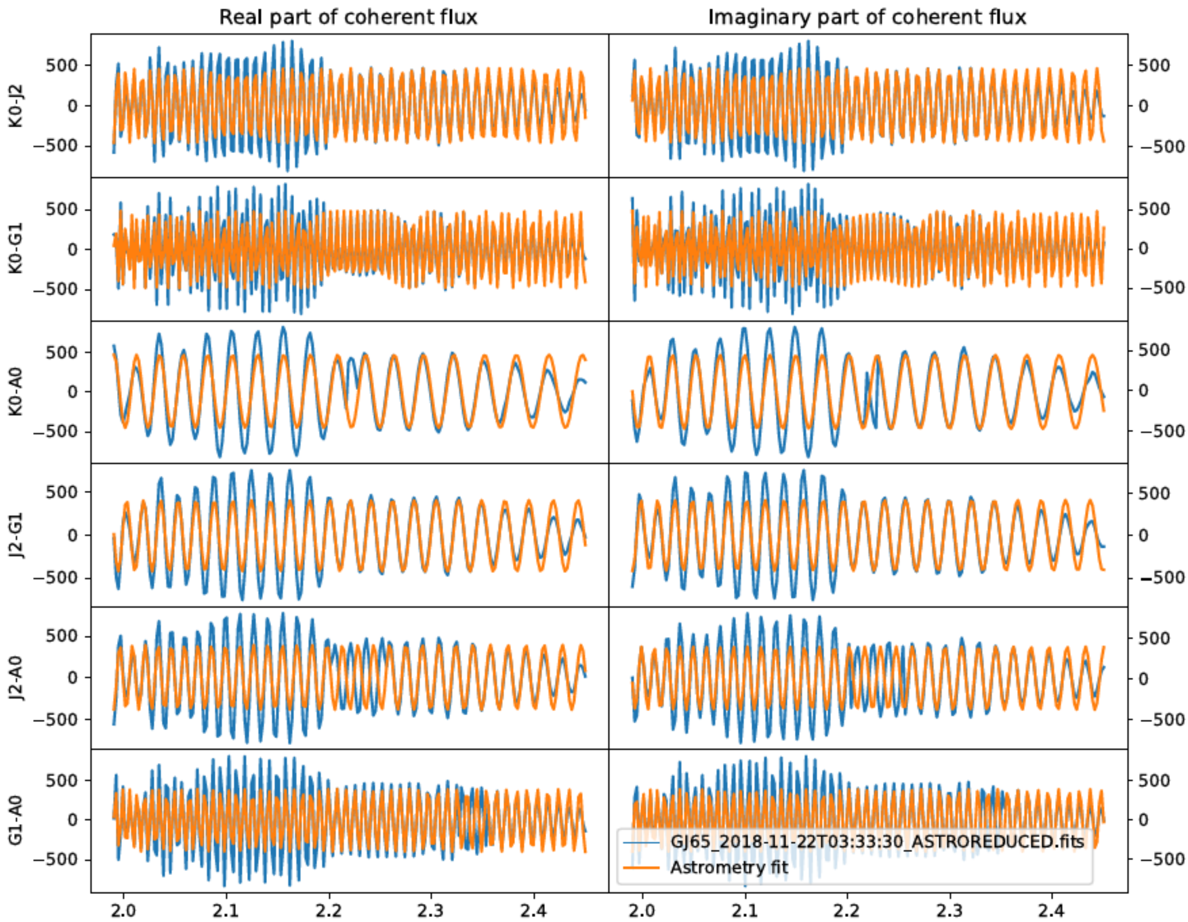


Figure 11.2: *Fits to the complex coherent flux to the six baselines for one of the observations.*

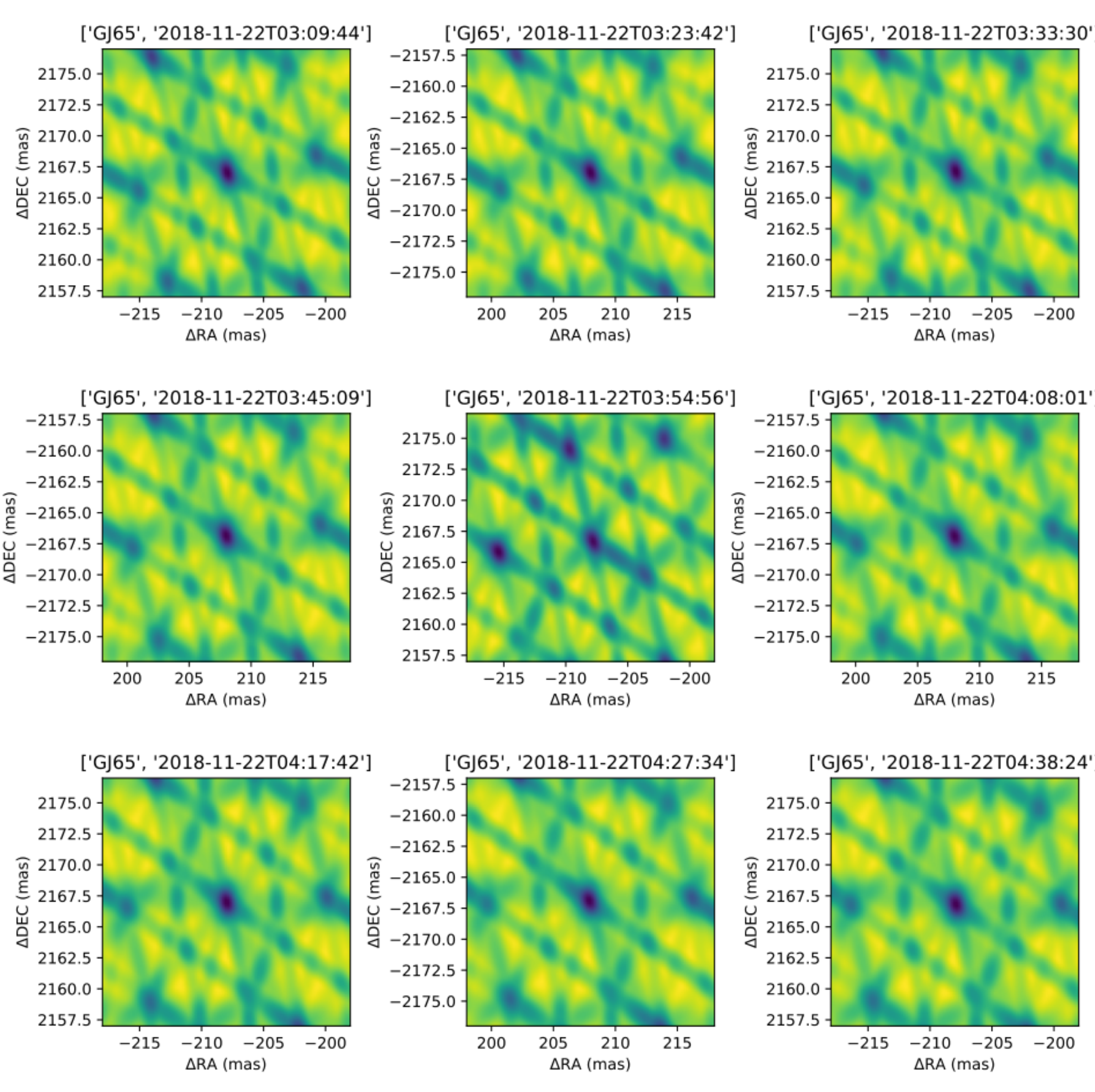


Figure 11.3: χ^2 *surfaces for each observation. The best astrometric solution is found in the center at RA = -208.73 and DEC = -2166.90 mas.*

11.6 Data files for planet observation

The list of raw files included in the astrometry demo data set is summarized in Table 11.2. This list was produced with the following command (part of the ESO SciSoft collection):

```
dfits *.fits | fitsort ins.sobj.name ft.robj.name ins.sobj.x ins.sobj.y OBJECT
```

These data need to be reduced with the pipeline, as described above, before using the Python tools for the astrometic analysis.

Table 11.2: The GRAVITY Reflex workflow astrometry tutorial data set, on-axis dual field mode data with medium spectral resolution and combined polarization. The units of the X and Y offsets are in milli-arcseconds.

File	SOBJ.NAME	FT.ROBJ.NAME	SOBJ.X	SOBJ.Y	OBJECT
GRAVI.2020-03-08T00:15:44.535.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T00:16:50.538.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T00:22:53.554.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T00:29:02.569.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T00:30:02.572.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T00:31:17.575.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T00:32:26.577.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T00:38:23.592.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T00:44:29.608.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T00:45:38.610.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T00:51:35.626.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T00:59:11.645.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T01:00:26.648.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T01:06:32.663.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T01:12:56.679.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T01:19:05.695.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T01:20:05.698.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T01:21:26.701.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T01:22:35.704.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T01:28:32.719.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T01:34:41.735.fits	SCI_Betpicc	BetaPictoris	0.000	0.000	betaPic
GRAVI.2020-03-08T01:35:47.738.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T01:41:47.753.fits	SCI_Betpicc	BetaPictoris	-72.072	-118.118	betaPic
GRAVI.2020-03-08T07:33:42.701.fits					DARK
GRAVI.2020-03-08T07:35:06.705.fits					FLAT
GRAVI.2020-03-08T07:35:48.707.fits					FLAT
GRAVI.2020-03-08T07:36:30.709.fits					FLAT
GRAVI.2020-03-08T07:37:12.710.fits					FLAT
GRAVI.2020-03-08T07:38:03.712.fits					WAVE,SC
GRAVI.2020-03-08T07:44:09.728.fits					WAVE
GRAVI.2020-03-08T07:49:24.741.fits					P2VM
GRAVI.2020-03-08T07:52:18.748.fits					P2VM
GRAVI.2020-03-08T07:55:12.756.fits					P2VM
GRAVI.2020-03-08T07:58:06.763.fits					P2VM
GRAVI.2020-03-08T08:01:00.770.fits					P2VM
GRAVI.2020-03-08T08:03:54.778.fits					P2VM
GRAVI.2020-03-08T08:07:03.786.fits					DARK
GRAVI.2020-03-08T08:08:27.789.fits					DARK

An example of the OB definition for such observations is shown in Fig. 11.4.

GRAVITY_dual_obs_exp	1	2	3	4	5	6
Science integration time (DI...	0.3	10	0.3	0.3	10	0.3
Number of science frames (...	64	32	64	64	32	64
Number of sky frames (NDIT)	64	32	64	64	32	64
Sequence of HWP offsets (d...	0.0	0.0	0.0	0.0	0.0	0.0
Sequence of observations O...	O	O O	O S	O	O O	O
Sequence of SC relative RA ...	0.072	-72.0	0.072	0.072	-72.0	0.072
Sequence of SC relative DE...	0.118	-118.0	0.118	0.118	-118.0	0.118
Sky dRA offset in milliarcsec...	2000.0	2000.0	2000.0	2000.0	2000.0	2000.0
Sky dDEC offset in milliarcse...	2000.0	2000.0	2000.0	2000.0	2000.0	2000.0
Data product category	SCIENCE	SCIENCE	SCIENCE	SCIENCE	SCIENCE	SCIENCE

Figure 11.4: *OB definition of a dual field on-axis observation.*

11.6.1 Analysis

As a first step, a configuration file needs to be created with information on the type of observation, the directory with pipeline products (files of type ASTROREDUCED, in OIFITS format), and on other miscellaneous parameters. It is assumed in the following that the directory *astroreduced* with the pipeline products is a sub-direcotry of the working directory. Furthermore, it is assumed that the directory *exogravity* is above the working directory. The following command creates a configuration file (`betapic.yml`):

```
python ../exogravity/create_config.py datadir="./astroreduced"
output="betapic.yml" ralim=\[-76,-68\] declim=\[-123,-115\]
suffix="ASTROREDUCED" calib_strategy="nearest"
```

For files with split polarimetric output, the OI_VIS extension can be either 11 or 12, otherwise it would be 10. Please ignore the fact that the extension number is not identical with the extension number listed in the OIFITS file. The messages issued by the script should state that files of type SWAP have been found, which is a standard observation sequence for dual field observations of two targets (FT and SC) which are both bright enough for fringe tracking.

The messages from the script should be like this:

```
nra not provided in args.  Default:  nra=100
ndec not provided in args.  Default:  ndec=100
nopd not provided in args.  Default:  to nopd=100
star_order not provided in args.  Default:  star_order=4
Value for noinv option not set.  Default:  noinv=False
Value for reflag not given.  Default:  reflag = True
Contrast file not given.  Constant contrast will be used
extension not given.  Using basic value '10'.
star_diameter not provided in args.  Default:  star_diameter=0 (point source)
corr_met not specified.  Using 'sylvestre'
corr_disp not specified.  Using 'sylvestre'
SWAP target name not given.  Assuming the observation to be on-axis (no swap)
reduction not given.  Using default 'astrored'
phaseref_arclength_threshold not given.  Using default value of 5
ft_flux_threshold not given.  Using default value of 0.2
By default, no baseline will be ignored.  Add baseline indices to 'ignore_baselin
in the yml to ignore some baselines.
4 files found
```

```
Loading ./astroreduced/SCI_BetaPictoris_2020-03-08T00:15:54_ASTROREDUCED.fits
Target is betaPic; Fiber distance is 0.00 mas.  Assuming file to be on star.
Loading ./astroreduced/SCI_BetaPictoris_2020-03-08T00:17:00_ASTROREDUCED.fits
Target is betaPic; Fiber distance is 138.37 mas.  Assuming file to be on planet.
Loading ./astroreduced/SCI_BetaPictoris_2020-03-08T00:23:02_ASTROREDUCED.fits
Target is betaPic; Fiber distance is 138.37 mas.  Assuming file to be on planet.
Loading ./astroreduced/SCI_BetaPictoris_2020-03-08T00:29:11_ASTROREDUCED.fits
Target is betaPic; Fiber distance is 0.00 mas.  Assuming file to be on star.
RA grid set to [-76.00, -68.00] with 100 points
DEC grid set to [-123.00, -115.00] with 100 points
No SWAP file detected.  Setting phaseref_mode to STAR.
Saved config for 4 files to betapic.yml
```

Once the configuration file has been created, the following python script is executed to create a phase reference from the star observations.

```
python ../../exogravity/create_phase_reference.py config_file=betapic.yml
```

```
Loading file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:17:00_ASTROREDUCED.fit
File is on planet.  FT coherent flux:  2.72e+06
Loading file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:23:02_ASTROREDUCED.fit
File is on planet.  FT coherent flux:  2.67e+06
Loading file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:29:11_ASTROREDUCED.fit
File is on star
Loading file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:15:54_ASTROREDUCED.fit
File is on star
A total of 233 points have been flagged in ./astroreduced/SCI_BetaPictoris_2020-0
(below FT threshold of 4.37e+05)
Creating the visibility reference from 2 star observations.
Saving reference visibility in ./astroreduced/SCI_BetaPictoris_2020-03-08T00:17:0
Saving reference visibility in ./astroreduced/SCI_BetaPictoris_2020-03-08T00:23:0
```

Finally, the following script is used determine the planet coordinates from their phases referenced to the stellar phase reference.

```
python ../../exogravity/astrometry_reduce.py config_file=betapic.yml figdir=figur
```

```
Loading file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:17:00_ASTROREDUCED.fit
File is on planet.  FT coherent flux:  2.72e+06
Loading file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:23:02_ASTROREDUCED.fit
File is on planet.  FT coherent flux:  2.67e+06
gofast flag is set.  Averaging over DITs
Averaging file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:17:00_ASTROREDUCED.f
Averaging file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:23:02_ASTROREDUCED.f
Retrieving visibility references from fits files
```

```
Subtracting phase reference to each planet OI.
Create projector matrices (p_matrices) (1/2)
Create projector matrices (p_matrices) (2/2)
Starting calculation of H matrices
Calculating H (1/2)
Calculating H (2/2)
Projecting visibilities (1/2)
Projecting visibilities (2/2)
Inverting covariance matrices (1/2)
Inverting covariance matrices (2/2)
RA grid:  [-76.00, -68.00] with 100 points
DEC grid:  [-123.00, -115.00] with 100 points
Calculating chi2Map for file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:17:00_
Calculating chi2Map for file ./astroreduced/SCI_BetaPictoris_2020-03-08T00:23:02_
Looking for local chi2 minimun for file ./astroreduced/SCI_BetaPictoris_2020-03-0
Looking for local chi2 minimun for file ./astroreduced/SCI_BetaPictoris_2020-03-0
Looking for best OPD on each baseline for file ./astroreduced/SCI_BetaPictoris_20
is different from global minimum
Looking for best OPD on each baseline for file ./astroreduced/SCI_BetaPictoris_20
is different from global minimum
RA: -71.84+-0.171 mas
DEC: -119.85+-0.114 mas
COV: -1.00
RA (from combined map):  -71.88+-0.171 mas
DEC (from combined map):  -119.85+-0.114 mas
Contrast obtained (mean, min, max):  4.11e-05, 3.95e-05, 4.28e-05
```

The script will deposit two files in the `figures` folder, one (FITS format) containing the astrometric χ^2 maps for each observation, and the other (PDF) showing the combined χ^2 map (Fig. 11.5) with the minimum shown in dark blue color).

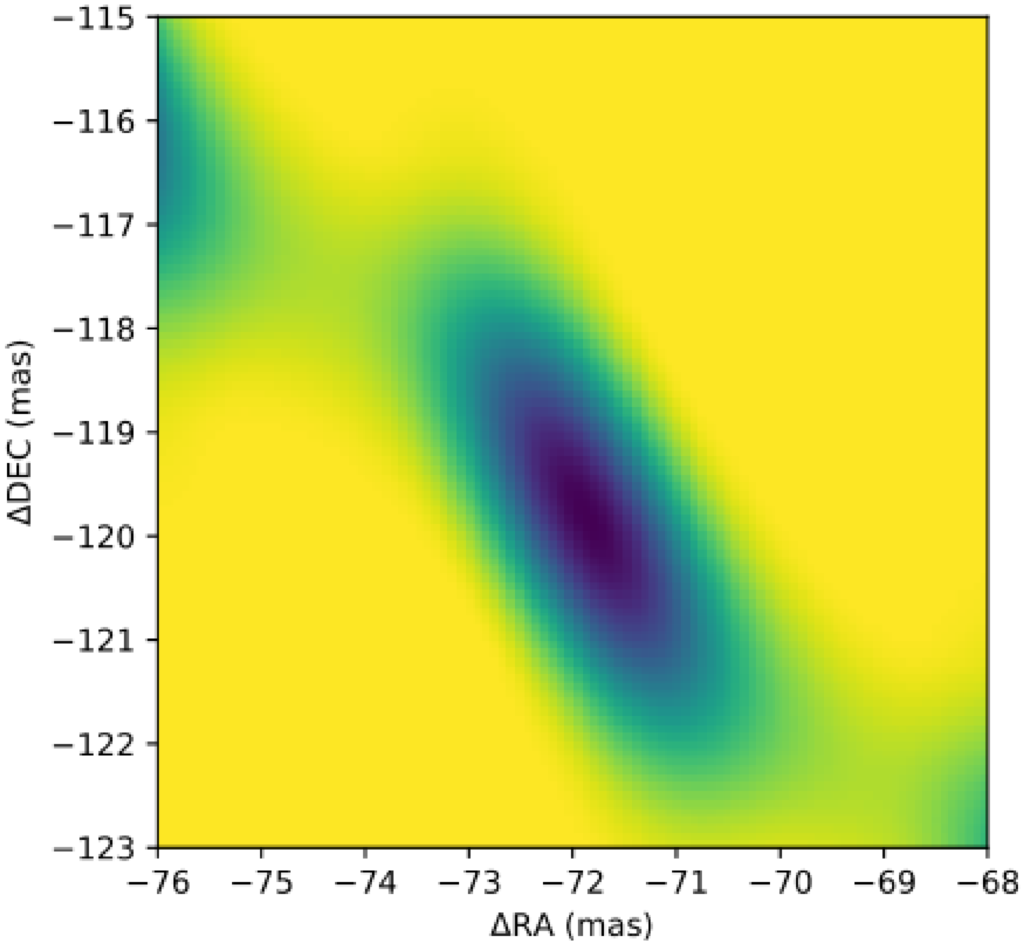


Figure 11.5: χ^2 *map of the on-axis observation (minimum shown in dark blue.*

12 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 3, 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`

13 Troubleshooting

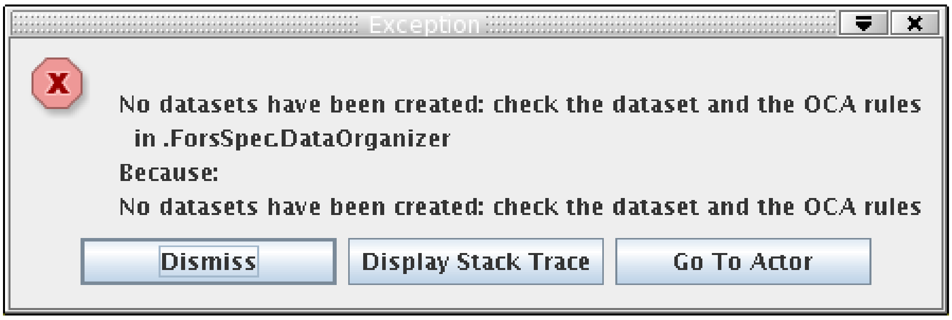


Figure 13.1: *TheDataOrganizer interactive window reports an error ":No DataSets have been created, check the data set and the OCA rules.".*

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 13.1.)**
   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. `DARK`). From this the user has then to determine the missing raw data (in this case dark frames). If static calibrations are missing the mechanism unfortunately does not work, but should be found by `reflex` in
   `<install_directory>/calib/<pipeline_version>/cal`

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

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

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

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

Acknowledgements

This tutorial has contributions from the GRAVITY IOT.

[1] G. Duvert, J. Young, and C.A. Hummel. OIFITS 2: the 2nd version of the Data Exchange Standard for Optical (Visible/IR) Interferometry. *Astronomy & Astrophysics*, 597:A8, 2015. 38

[2] GRAVITY IOT GRAVITY Consortium. *GRAVITY 2016 User Manual*, 2016. VLT-MAN-ESO-XXXXX-XXX. 7

[3] P. H. Hauschildt, F. Allard, and E. Baron. The nextgen model atmosphere grid. *The Astrophysical Journal*, 512:377–385, 1999. 36

[4] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, *https://www.eso.org/reflex/*, 0.7 edition, 2012. VLT-MAN-ESO-19000-5037. 16, 54, 55