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

MIDI Pipeline User Manual

VLT-MAN-ESO-19500-3310

Issue 2.9.5

Date 2023-05-19

	Name	Date	Signature
Prepared:	ESO MIDI Pipeline Team	2023-05-19	
Approved:	P.Ballester, M. Peron		
Released:			

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.1.0	21/04/2004	All	Draft version
1.5.4	01/02/2006	All	Reflects midi-1.5.4
1.5.9	01/07/2006	All	Reflects midi-1.5.9
1.6.2	10/11/2006	All	Reflects midi-1.6.2
2.6.0	01/04/2010	All	Reflects midi-2.6.0
2.7.0	01/04/2011	4, 8.1, 5, 10.5	Reflects midi-2.7.0
2.8.1	01/04/2012	4, 10.6	Reflects midi-2.8.1
2.8.2	01/04/2013	4	Reflects midi-2.8.2
2.8.3	01/04/2014	4	Reflects midi-2.8.3
2.8.5	01/04/2016	4	Reflects midi-2.8.5
2.9.2	01/04/2019	4	Reflects midi-2.9.2
2.9.5	19/05/2023	4	Reflects midi-2.9.5

This page was intentionally left blank

Contents

1 Introduction **9**

1.1 Purpose . . . 9

1.2 Acknowledgment . . . 9

1.3 Scope . . . 10

1.4 Reference Documents . . . 10

2 Overview **12**

3 MIDI Instrument Description **13**

4 What's new **14**

4.1 What's new in pipeline release 2.9.5 . . . 14

4.2 What's new in pipeline release 2.9.2 . . . 14

4.3 What's new in pipeline release 2.8.5 . . . 14

4.4 What's new in pipeline release 2.8.3 . . . 14

4.5 What's new in pipeline release 2.8.2 . . . 14

4.6 What's new in pipeline release 2.8.1 . . . 14

4.7 What's new in pipeline release 2.7.0 . . . 15

5 Quick Start **16**

5.1 An introduction to Gasgano and Esorex . . . 16

5.1.1 Using Gasgano . . . 16

5.1.2 Using Esorex . . . 20

5.2 Example of HIGH_SENS data reduction using Esorex . . . 22

5.3 Example of SCI_PHOT data reduction using Esorex . . . 25

6 Known Problems **31**

7 Instrument Data Description **32**

8 Static Calibration Data **36**

8.1 Calibrator Database . . . 36

8.2 Wavelength Calibration Database . . . 36

9 Data Reduction **37**

9.1 Data reduction overview . . . 37

9.2 Reduction Cascade . . . 37

10 Pipeline Recipes Interfaces **39**

10.1 midi_profile . . . 39

10.1.1 Input files . . . 39

10.1.2 Output files . . . 39

10.1.3 Recipe parameters . . . 40

10.1.4 Quality control parameters . . . 40

10.2 midi_kappamatrix . . . 41

10.2.1 Input files . . . 41

10.2.2 Output files . . . 41

10.2.3 Recipe parameters . . . 42

10.2.4 Quality control parameters . . . 42

10.3 midi_raw_to_fitsimage . . . 43

10.3.1 Input files . . . 43

10.3.2 Output files . . . 43

10.3.3 Recipe parameters . . . 44

10.4 midi_intopd . . . 45

10.4.1 Input files . . . 45

10.4.2 Output files . . . 45

10.4.3 Recipe parameters . . . 45

10.4.4 Quality control parameters . . . 46

10.5 midi_fringe_all . . . 47

10.5.1 Input for HIGH_SENS data . . . 47

10.5.2 Input for SCI_PHOT data . . . 47

10.5.3 Output files . . . 48

10.5.4 Recipe parameters . . . 49

10.5.5 Quality control parameters . . . 49
10.6 midi_acq . . . 50
10.6.1 Input files . . . 50
10.6.2 Output files . . . 50
10.6.3 Quality control parameters . . . 50
10.7 midi_wavecal . . . 51
10.7.1 Input files . . . 51
10.7.2 Output files . . . 52
10.7.3 Quality control parameters . . . 52
10.8 midi_detlin . . . 53
10.8.1 Input files . . . 53
10.8.2 Output files . . . 53
10.8.3 Quality control parameters . . . 53
10.9 midi_detron . . . 54
10.9.1 Input files . . . 54
10.9.2 Output files . . . 54
10.9.3 Quality control parameters . . . 54
10.10 midi_dsptrn . . . 55
10.10.1 Input files . . . 55
10.10.2 Output files . . . 56
10.10.3 Quality control parameters . . . 56
10.11 midi_refpix . . . 57
10.11.1 Input files . . . 57
10.11.2 Output files . . . 57
10.11.3 Quality control parameters . . . 57

11 Algorithms **58**
11.1 midi_profile . . . 58
11.2 midi_kappamatrix . . . 60
11.3 midi_intopd . . . 62
11.4 midi_fringe_all: HIGHSENS observations . . . 64

11.4.1 Data Compression . . . 64
11.4.2 Photometry computation . . . 64
11.4.3 Fringe Estimation . . . 66
11.4.4 Visibility Computation . . . 66
11.4.5 Transfer Function and Calibrated Visibility Computation (optional) . . . 68
11.5 midi_fringe_all: SCIPHOT observations . . . 69
11.6 midi_acq . . . 70
11.7 midi_wavecal . . . 71
11.8 midi_detlin . . . 72
11.9 midi_detron . . . 73
11.10 midi_dsptrn . . . 74
11.11 midi_refpix . . . 76

A Installation **77**

B Abbreviations And Acronyms **78**
B.1 Abbreviations . . . 78
B.2 Glossary . . . 79

C QC1 Dictionary **81**
C.1 QC parameter of the midi_profile recipe . . . 81
C.2 QC parameter of the midi_kappamatrix recipe . . . 81
C.3 QC parameter of the midi_intopd recipe . . . 82
C.4 QC parameter of the midi_fringe_all recipe . . . 82
C.5 QC parameter of the midi_acq recipe . . . 85
C.6 QC parameter of the midi_detlin recipe . . . 86
C.7 QC parameter of the midi_detron recipe . . . 86
C.8 QC parameter of the midi_dsptrn recipe . . . 86
C.9 QC parameter of the midi_refpix recipe . . . 88
C.10 QC parameter of the midi_wavecal recipe . . . 90

D Troubleshooting Guide **91**

1 Introduction

1.1 Purpose

The MIDI pipeline is a subsystem of the *VLT Data Flow System* (DFS). It is used in two operational environments, for the ESO *Data Management Operations* (DMO), and for the *Paranal Science Operations* (PSO), in the quick-look assessment of data, in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. Additionally, the MIDI pipeline recipes are made public to the user community, to enable a more personalized processing of the data from the instrument.

1.2 Acknowledgment

The following persons and organizations contributed to the MIDI instrument and pipeline:

The MIDI consortium who built and commissioned MIDI consists of the following European institutes: Max Planck Institut fuer Astronomie, Heidelberg, Germany. Netherlands Graduate School for Astronomy (NOVA), Leiden, Netherlands. Department of Astronomy, Leiden Observatory, Netherlands. Kapteyn Astronomical Institute, Groningen, Netherlands. Astronomical Institute, Utrecht University, Netherlands. Astronomical Institute, University of Amsterdam, Netherlands. Netherlands Foundation for Research in Astronomy, Dwingeloo, Netherlands. Space Research Organization Netherland, Utrecht and Groningen, Netherlands. Thueringer Landessternwarte, Tautenburg, Germany. Kiepenheuer-Institut fuer Sonnenphysik, Freiburg, Germany. Observatoire de Paris-Meudon, Meudon, France. Observatoire de la Cote d'Azur, Nice, France.

Main contributors have been the Netherlands Graduate School for Astronomy (NOVA) in Leiden, Netherlands and the Max Planck Institut fuer Astronomie, Heidelberg, Germany.

We would also like to thank all users as they provided very valuable comments, either indicating bugs, limitations, or giving suggestions to improve the pipeline.

Finally the Pipeline Development Team included at least the following people:

- Pascal Ballester: Coordinated and contributed in all aspects of the MIDI pipeline development
- Isabelle Percheron: Evaluated the MIDI pipeline in Garching and made numerous contributions and useful comments during the development of the pipeline
- Sebastien Morel: Evaluated the MIDI pipeline in Paranal and made numerous contributions in the development of the pipeline
- Jeff Meisner: Developed the major part of the algorithm for the visibility calculation
- Cyrus Sabet: MIDI Pipeline developer
- Armin Gabasch: MIDI Pipeline developer

1.3 Scope

This document describes the MIDI pipeline.

For further details on various aspects of the DFS operation please refer to [1], [2] and [3]. For Up-to-data information and documentation on the MIDI instrument reference should be made to [4], [5] and [6]

1.4 Reference Documents

[1] ESO. *VLT Data Flow System Operations Model for VLT/VLTI Instrumentation*. VLT-PLA-ESO-19000-1183. 10

[2] ESO. *VLT Data Flow System Specifications for Pipeline and Quality Control*. VLT-SPE-ESO-19600-1233. 10

[3] ESO. *Deliverables Specification*. VLT-SPE-ESO-19000-1618 (2.0). 10

[4] ESO. *VLT MIDI User Manual (Version 77.1, 31.Aug.05)*. VLT-MAN-ESO-15820-3519 . 10, 13

[5] ESO. *VLT MIDI Template Manual. (Version 76.1, 30.May.05)*. VLT-MAN-ESO-15820-3520. 10

[6] ESO. *VLTI Data Interface Control Document (Issue 1.0 03.Jun.02)*. VLT-SPE-ESO-15000-2764. 10, 13, 32

[7] ESO. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 12, 17

[8] *ESOREX home page*. http://www.eso.org/cpl/esorex.html. 12, 16, 20, 21, 32

[9] *GASGANO home page*. http://www.eso.org/gasgano. 12, 16, 17, 32

[10] P. Kervella, D. Ségransan, and V. Coudé du Foresto. Data reduction methods for single-mode optical interferometry. Application to the VLTI two-telescopes beam combiner VINCI. *A&A*, 425:1161–1174, October 2004. 37, 41, 60

[11] J. D. Scargle. Studies in astronomical time series analysis. II - Statistical aspects of spectral analysis of unevenly spaced data. *ApJ*, 263:835–853, December 1982. 37, 45, 63

[12] K. Horne. An optimal extraction algorithm for CCD spectroscopy. *PASP*, 98:609–617, June 1986. 41, 60, 61

[13] P. R. Lawson. Group-delay tracking in optical stellar interferometry with the fast Fourier transform. *Journal of the Optical Society of America A*, 12:366–374, February 1995. 45, 62

2 Overview

In collaboration with instrument consortia, the Data Flow Systems Department (DFS) of the Data Management and Operation Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

Data quality control: pipelines are used to produce the quantitative information necessary to monitor instrument performance.

Master calibration product creation: pipelines are used to produce master calibration products (e.g., wavelength dispersion solutions, ...).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes (e.g., optimally extracted spectra, calibrated visibilities, ...). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

Instrument pipelines consist of a set of data processing modules that can be called from opportune front-end applications, such as the automatic data management tools available on Paranal.

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* [7] and *Esorex* [8], both included in the pipeline distribution. These applications can also be downloaded separately from the ESO web pages (see [9] and [8]). An illustrated introduction to Gasgano is provided in the "Quick Start" Section of this manual (see page 16).

The MIDI instrument and the different types of MIDI raw frames and auxiliary data are described in sections 3, 7, and 8.

A brief introduction to the usage of the available reduction recipes is presented in section 5. In section 6 we advise the user of any known data reduction problems as well as their possible solutions.

An overview of the data reduction, the interfaces and the objectives of each recipe is provided in section 9.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in Section 10. The algorithms involved in the above recipes are described in section 11.

In Appendix A the installation of the MIDI pipeline recipes is described and in Appendix B a list of used abbreviations and acronyms is given.

3 MIDI Instrument Description

MIDI (see also Fig.3.0.0) belongs to the first generation of VLTI instruments. It is a mid-infrared interferometric instrument covering the N band from approximately 8 μm to 13 μm. It is the first available scientific VLTI instrument. MIDI is a single-mode instrument and produces up to four outputs in order to monitor the fluctuations of phase due to turbulence. Two outputs represent the data for the interferometric signals and the other two provide the data for the photometric signals.

Conceptual studies for a VLTI mid-infrared instrument started in 1997. The Final Design Review of MIDI was passed in early 2000 for the hardware, and mid-2001 for the software. The integration took place at the Max-Planck Institut fuer Astronomie, Heidelberg, Germany. After Preliminary Acceptance in Europe in September 2002, MIDI was shipped to Paranal and re-assembled there in November 2002, where it obtained its first fringes with the UTs on the 12 December 2002.

During the year 2003, intensive commissioning and paranalization transformed MIDI into a premium scientific instrument that can now be offered to the worldwide community of astronomers.

Inside MIDI, the beam combination is achieved in a plane close to the re-imaged pupil, and the signal is detected in an image plane (infinity). From the beam combiner onwards, the two interfering beams have a common optical axis, both for the reflected and the transmitted outputs. MIDI in the interferometric laboratory consists of two main parts; the warm optics on the MIDI table and the cold optics in cryostat. In addition, in the adjacent Combined Coude Laboratory, there are an infrared CO2 laser (used for calibration measurements), the control electronics and the cooling system. MIDI uses the cold optics for the beam combiner because radiation at 10 μm is dominated by thermal emission of the environment. Baffling within the instrument is mainly done by a cold pupil stop just downstream to the entrance window of the cryostat. OPD modulation is done in the warm laboratory environment.
For further details on the optical layout please refer to [4, 6].

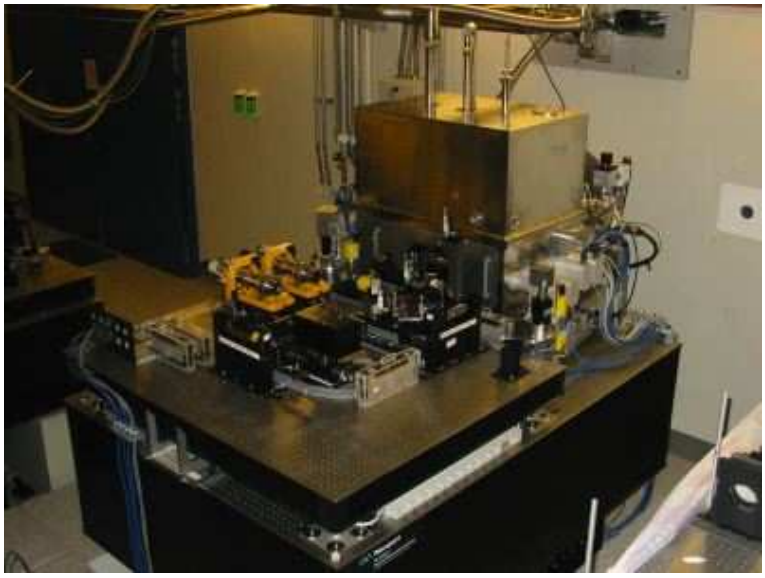

Figure 3.0.0: *MIDI instrument on Paranal*

4 What's new

4.1 What's new in pipeline release 2.9.5

- The pipeline has been updated to work with cpl version 7.3.
- The pipeline installation routines were standardized and updated.

4.2 What's new in pipeline release 2.9.2

- The pipeline has been updated to work with cpl version 7.1.1.
- The pipeline installation routines were standardized and updated.

4.3 What's new in pipeline release 2.8.5

- The pipeline has been updated to work with cpl version 7.
- The pipeline installation routines were standardized and updated.

4.4 What's new in pipeline release 2.8.3

- The pipeline has been updated to the new cpl version 6.4 and the build system has been improved.

4.5 What's new in pipeline release 2.8.2

- The pipeline has been updated to the new cpl version 6.3 and the build system, based on the autotools, has been modified in order to build on a variety of different platforms.

4.6 What's new in pipeline release 2.8.1

- The midi_acq recipe derives two new products (PRO.CATG MIDI_ACQ_FOV_DATA1 and MIDI_ACQ_FOV_DATA2) together with new QC parameter related to the field of view of Midi:

```
QC GAUSS FIT X
QC GAUSS FIT Y
QC GAUSS FIT SIGMA X
QC GAUSS FIT SIGMA Y
QC GAUSS MEASURED FWHM AT X
QC GAUSS MEASURED FWHM AT Y
QC CENTROID X
QC CENTROID Y
QC CENTROID MEASURED FWHM AT X
QC CENTROID MEASURED FWHM AT Y
```

- New calibrator-list:

```
FILE                          PRO.CATG
calibrator_database_N.fits    CALIB_DATABASE_N
```

- The number of required transitions (e.g. target->sky) has been lowered from 10 to 3, i.e. if a Midi file has more then 3 transitions, the recipe will reduce the file.
- The QC parameter `QC BL LENGTH` and `QC BL ANGLE` derived by the midi_fringe_all recipe are always written to the product header, even if a calibrator could not be identified in the calibrator database.

4.7 What's new in pipeline release 2.7.0

- A new recipe parameter "checkSof" has been added to the midi_fringe_all recipe (default value set to TRUE). If deactivated (checkSof=FALSE) the pipeline will not check if all rawfiles in the SOF have the same TPL START or a continuous OCS TPL FILENO header keyword. This allows expert users to combine and process files coming from different templates. USE WITH CAUTION!!
- An additional midi_fringe_all recipe product named MIDI_CorrelatedFlux.fits with the PRO.CATG CORRELATED_FLUX has been added. The product contains a table with the wavelength, the correlated flux, the error of the correlated flux and a flag (flag equals to zero if the corresponding channel could be properly reduced). Please note that this new product is still under evaluation.
- A new calibrator database has been included. Moreover, the internal format has been changed to homogenize the Midi and Amber calibrator database. The old calibrator database file can not been used anymore!

5 Quick Start

This section describes the most immediate usage of the MIDI pipeline recipes. For a complete list of the available recipes, please see Section 9, page 37.

5.1 An introduction to Gasgano and Esorex

Before being able to apply pipeline recipes to a set of data, the data must be opportunely classified, and associated with the appropriate calibrations. The *Data Classification* consists of tasks such as: "What kind of data am I?", e.g., PHOTOMETRY, "To which group do I belong?", e.g., to a particular Observation Block or template. *Data Association* is the process of selecting appropriate calibration data for the reduction of a set of raw science frames. Typically, a set of frames can be associated if they share a number of properties, such as instrument and detector configuration. As all the required information is stored in the FITS headers, data association is based on a set of keywords (called "association keywords") and is specific to each type of calibration.

The process of data classification and association is known as data organization. The *DO Category* is the label assigned to a data type as a result of data classification.

An instrument pipeline consists of a set of data processing modules that can be called from different host applications, either from the command line with *Esorex* [8], from the automatic data management tools available at Paranal, or from the graphical *Gasgano* tool [9].

Gasgano is a data management tool that simplifies the data organization process, offering automatic data classification and making the data association easier (*even if automatic association of frames to a recipe is not yet provided*). *Gasgano* determines the classification of a file by applying an instrument specific rule, while users must provide this information to the recipes when they are executed manually using *Esorex* from the command line. In addition, *Gasgano* allows the user to execute directly the pipeline recipes on a set of selected files.

5.1.1 Using Gasgano

To get familiar with the MIDI pipeline recipes and their usage, it is advisable to begin with *Gasgano*, because it provides a complete graphic interface for data browsing, classification and association, and offers several other utilities such as easy access to recipes documentation and preferred data display tools.

Gasgano can be started from the system prompt in the following way:

```
gasgano &
```

The *Gasgano* main window will appear.

On Figure 5.1.1 (page 17), a view on a set of MIDI photometry data is shown as an example. *Gasgano* can be pointed to the directories where the data to be handled are located using the navigation panels accessible via the *Add/Remove Files* entry of the *File* menu (shown on the upper left of the figure).

The data are hierarchically organized as preferred by the user. After each file name the classification (*Do Category*) and many other file-related information are shown.

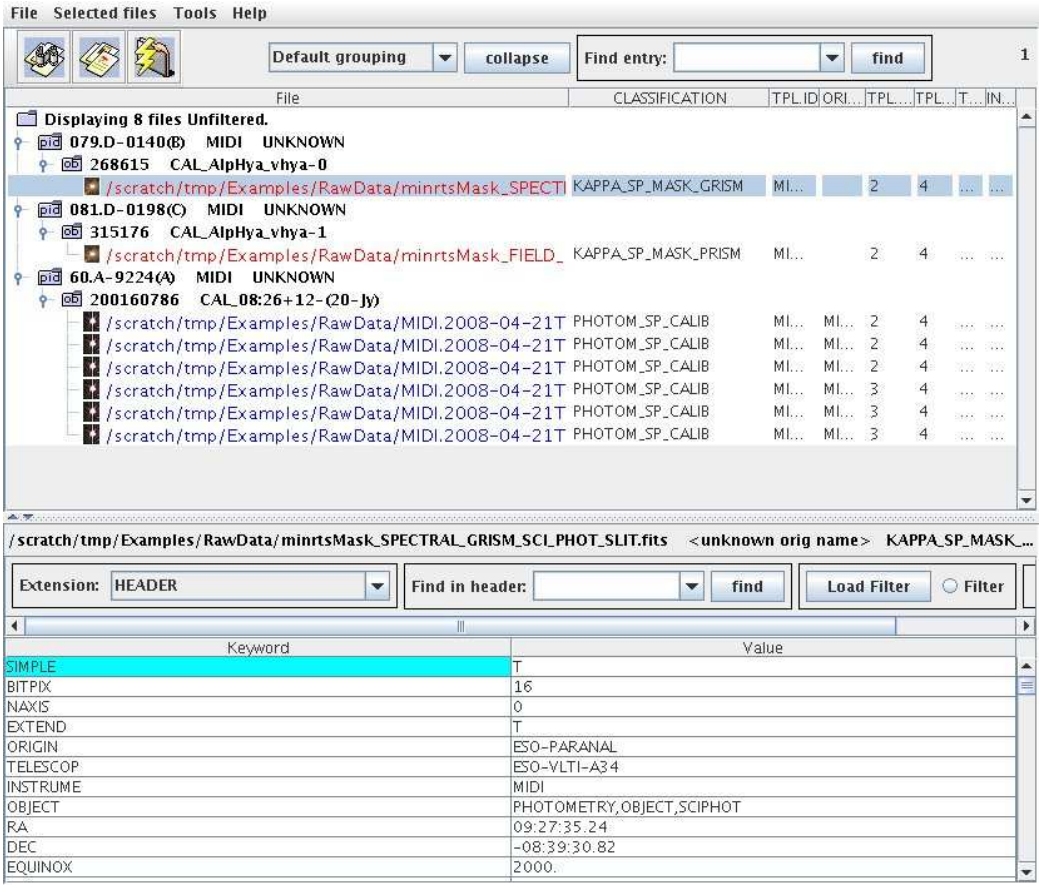

Figure 5.1.1: *The Gasgano main window.*

More information about a single frame can be obtained by clicking on its name: the corresponding FITS file header will be displayed on the bottom panel, where specific keywords can be opportunely filtered and searched. Images and tables may be easily displayed using the viewers specified in the appropriate *Preferences* fields.

Frames can be selected from the main window for being processed by the appropriate recipe: on Figure 5.1.2, photometry files together with the corresponding mask file are all selected and sent to the *midi_kappamatrix* recipe. This will open a *Gasgano* recipe execution window (see Figure 5.1.3), having all the specified files listed in its *Input Frames* panel.

Help about the recipe may be obtained from the *Help* menu. Before launching the recipe, its configuration may be opportunely modified on the *Parameters* panel (on top). The window contents might be saved for later use by selecting the *Save Current Settings* entry from the *File* menu, as shown in figure.

At this point the recipe can be launched by pressing the *Execute* button. Messages from the running recipe will appear on the *Log Messages* panel at bottom, and in case of successful completion the products will be listed on the *Output Frames* panel, where they can be easily viewed and located back on the Gasgano main window.

Please refer to the *Gasgano User's Manual* [7] for a more complete description of the *Gasgano* interface. See also [9].

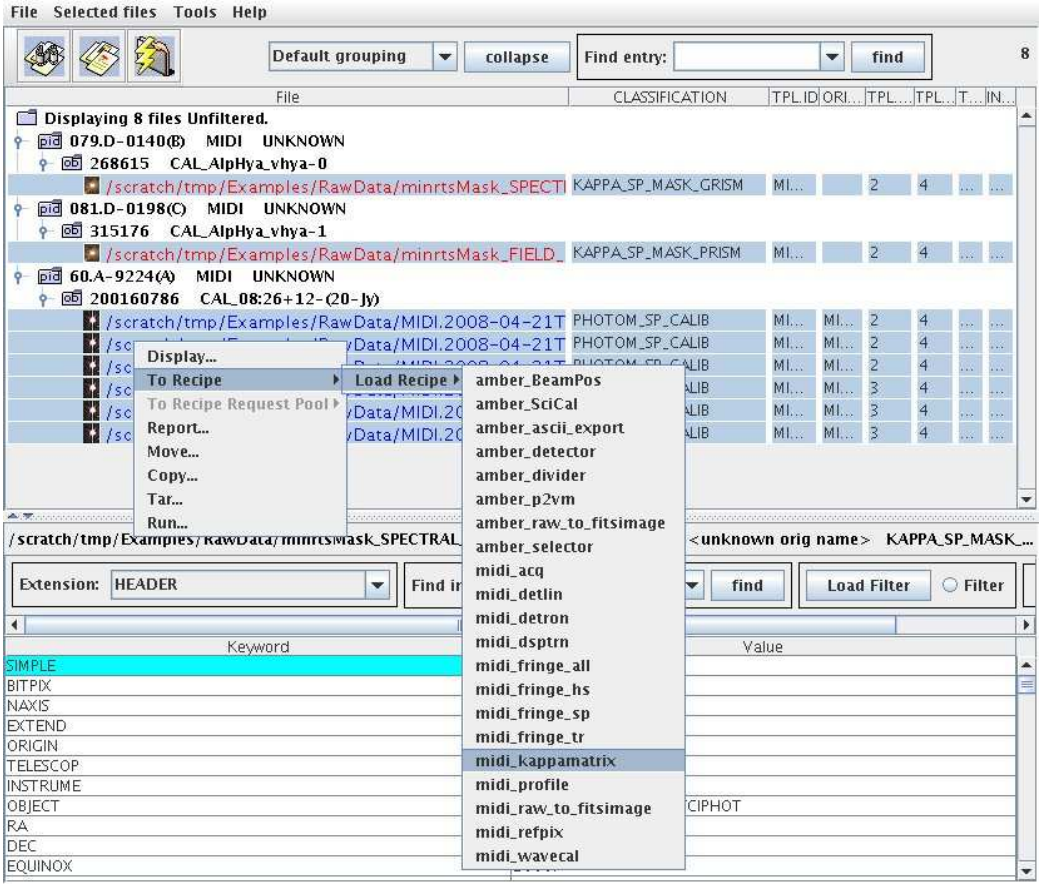

Figure 5.1.2: *Selecting files to be processed by a MIDI pipeline recipe.*

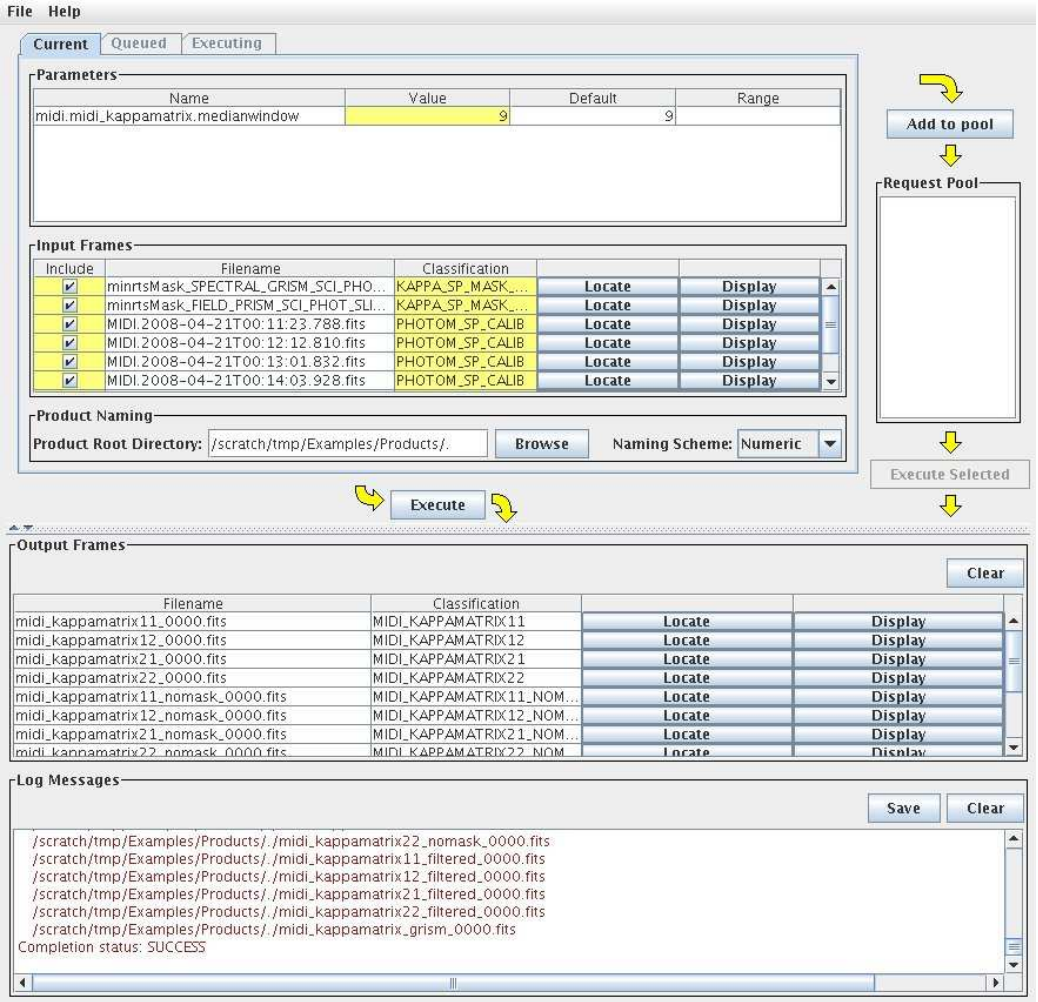

Figure 5.1.3: *The Gasgano recipe execution window.*

5.1.2 Using Esorex

Esorex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. On the other side, *Esorex* does not offer all the facilities available with *Gasgano*, and the user must classify and associate the data using the information contained in the FITS header keywords. The user should also take care of defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

The set-of-frames (SOFs): Each pipeline recipe is run on a set of input FITS data files. When using *Esorex* the filenames must be listed together with their DO category in an ASCII file, the *set-of-frames*, that is required when launching a recipe. [1]

Here is an example of SOF, valid for the *midi_kappamatrix* recipe:

```
2008-04-21/MIDI.2008-04-21T00:11:23.788.fits       PHOTOM_SP_CALIB
2008-04-21/MIDI.2008-04-21T00:12:12.810.fits       PHOTOM_SP_CALIB
2008-04-21/MIDI.2008-04-21T00:13:01.832.fits       PHOTOM_SP_CALIB
2008-04-21/MIDI.2008-04-21T00:14:03.928.fits       PHOTOM_SP_CALIB
2008-04-21/MIDI.2008-04-21T00:14:52.951.fits       PHOTOM_SP_CALIB
2008-04-21/MIDI.2008-04-21T00:15:41.973.fits       PHOTOM_SP_CALIB
calib/minrtsMask_FIELD_PRISM_SCI_PHOT_SLIT.fits    KAPPA_SP_MASK_PRISM
calib/minrtsMask_SPECTRAL_GRISM_SCI_PHOT_SLIT.fits KAPPA_SP_MASK_GRISM
```

This file contains the name of each input frame, and its DO category. The launched pipeline recipe will access the listed files when required by the reduction algorithm.

Note that the MIDI pipeline recipes do not verify in any way the correctness of the *DO Category* specified by the user in the SOF. The reason of this lack of control is that the MIDI recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 5.1.1, page 16).

A recipe handling an incorrect SOF may stop or display unclear error messages at best. In the worst cases, the recipe would apparently run without any problem, producing results that may look reasonable, but are actually flawed.

Esorex syntax: The basic syntax to use *Esorex* is the following:

```
esorex [esorex_options] recipe_name [recipe_options] set_of_frames
```

To get more information on how to customize *Esorex* (see also [8]) run the commands:

```
esorex --man
esorex --help
esorex --par
```

[1] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 5.1.3, page 19).

To generate a configuration file esorex.rc in the directory $HOME/.esorex run the command:

```
esorex --create-config
```

A list of all available recipes, each with a one-line description, can be obtained using the command:

```
esorex --recipes
```

All recipe parameters (aliases) and their default values can be displayed by the command

```
esorex --params recipe_name
```

To get a brief description of each parameter meaning execute the command:

```
esorex --help recipe_name
```

To get more details about the given recipe use the commands:

```
esorex --man recipe_name
esorex --help recipe_name
esorex --par recipe_name
```

Recipe configuration: Each pipeline recipe may be assigned an *Esorex* configuration file, containing the default values of the parameters related to that recipe.[2] The configuration files are normally generated in the directory `$HOME/.esorex`, and have the same name as the recipe to which they are related, with the file-name extension `.rc`. For instance, the recipe *midi_kappamatrix* has its *Esorex* default configuration file named `midi_kappamatrix.rc`, generated with the command:

```
esorex --create-config midi_kappamatrix
```

If a number of recipe parameters are specified on the command line, the given values will be used in the created configuration file.

A description of the recipe parameters is provided in Section 10, page 39.

Recipe execution: A recipe can be run by specifying its name to *Esorex*, together with the name of a set-of-frames. For instance, the following command line would be used to run the recipe *midi_kappamatrix* for processing the files specified in the set-of-frames `cal.sof`:

```
esorex midi_kappamatrix cal.sof
```

The recipe parameters can be modified either by editing directly the used configuration file, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name and before the SOF name, and they will supersede the system defaults and/or the configuration file settings.

For more information on *Esorex*, see [8].

[2] The *Esorex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 5.1.3, page 19).

5.2 Example of HIGH_SENS data reduction using Esorex

In the following, a typical MIDI reduction procedure for data taken in HIGH_SENS mode is described.[3] It is assumed that the following data are available:

Photometric exposures:

```
MIDI.2007-06-30T07:14:25.250.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:15:39.735.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:17:15.250.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:18:29.756.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:29:57.750.fits  PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:31:12.236.fits  PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:32:50.250.fits  PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:34:04.736.fits  PHOTOM_HS_SCIENCE
```

Interferometric exposures of a calibrator target:

```
MIDI.2007-06-30T07:09:46.000.fits  HIGH_SENS_CALIB
MIDI.2007-06-30T07:11:00.507.fits  HIGH_SENS_CALIB
MIDI.2007-06-30T07:12:15.013.fits  HIGH_SENS_CALIB
```

Interferometric exposures of a science target:

```
MIDI.2007-06-30T07:25:24.000.fits  HIGH_SENS_SCIENCE
MIDI.2007-06-30T07:26:38.507.fits  HIGH_SENS_SCIENCE
MIDI.2007-06-30T07:27:53.013.fits  HIGH_SENS_SCIENCE
```

File containing fundamental data of various calibrators. This file is delivered together with the pipeline:

```
calibrator_database_N.fits        CALIB_DATABASE_N
```

In the following, it is also assumed for simplicity that, in the *Esorex* configuration file, the flag *suppress-prefix* is set to `TRUE` and all raw files are located in the current directory. Moreover it is also assumed that the following directories are created in order to store the results of the different recipes:

```
Products_mask
Products_calib1
Products_science1_calibrated
```

As a first step a suitable mask/profile file has to be generated by using the recipe *midi_profile*.

The input SOF may be defined as follows:

File: *hs_mask1.sof*

[3] The procedure using *Gasgano* is conceptually identical.

```
MIDI.2007-06-30T07:14:25.250.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:15:39.735.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:17:15.250.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:18:29.756.fits  PHOTOM_HS_CALIB
MIDI.2007-06-30T07:29:57.750.fits  PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:31:12.236.fits  PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:32:50.250.fits  PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:34:04.736.fits  PHOTOM_HS_SCIENCE
```

This is the list of the raw frames used to derive the 2D profile of the spectral dispersed signal. Following the optimal extraction technique the product is used in various recipes to extract the signal with an optimal SNR. Moreover the product of this recipe also masks scattered light from the delay line tunnel when extracting the signal.

The following command line can now be given at the shell prompt:

```
esorex --output-dir=Products_mask midi_profile hs_mask1.sof
```

One main product and several diagnostic files (mainly for check purposes) are created in `Products_mask`:

midi_profile_prism.fits Main recipe product containing the profile of the signal

image_AOPEN_DATA1_profile.fits profile in the DATA1 section with telescope A open
image_AOPEN_DATA1_signal.fits signal in the DATA1 section with telescope A open
image_AOPEN_DATA2_profile.fits profile in the DATA2 section with telescope A open
image_AOPEN_DATA2_signal.fits signal in the DATA2 section with telescope A open
image_BOPEN_DATA1_profile.fits profile in the DATA1 section with telescope B open
image_BOPEN_DATA1_signal.fits signal in the DATA1 section with telescope B open
image_BOPEN_DATA2_profile.fits profile in the DATA2 section with telescope B open
image_BOPEN_DATA2_signal.fits signal in the DATA2 section with telescope B open

As you can see from the SOF we combined the photometry files of the calibrator and the science target in order to improve the SNR of the product. This should be done only after running the recipe on the calib and science data separately and checking (by e.g. blinking the diagnostic files) that the profile is not moving between the different observations. Moreover, one can fine-tune the result by changing the default recipe option `--threshold`. This threshold is used by the recipe to discriminate between source and background pixels (in sigma units). A typical result of this recipe is shown in Fig. 11.1.1.

The second step in the data reduction cascade is to reduce the calibrator observations using the recipe *midi_fringe_all*.

The input SOF may be defined as follows:

File: *hs_calib1.sof*

```
MIDI.2007-06-30T07:09:46.000.fits     HIGH_SENS_CALIB
MIDI.2007-06-30T07:11:00.507.fits     HIGH_SENS_CALIB
MIDI.2007-06-30T07:12:15.013.fits     HIGH_SENS_CALIB
MIDI.2007-06-30T07:14:25.250.fits     PHOTOM_HS_CALIB
MIDI.2007-06-30T07:15:39.735.fits     PHOTOM_HS_CALIB
MIDI.2007-06-30T07:17:15.250.fits     PHOTOM_HS_CALIB
MIDI.2007-06-30T07:18:29.756.fits     PHOTOM_HS_CALIB
Products_mask/midi_profile_prism.fits KAPPA_HS_MASK_PRISM
calibrator_database_N.fits              CALIB_DATABASE_N
```

In this SOF the `calibrator_database_N.fits` file is used to derive the transfer function to calibrate a science observation. If this file is not in the SOF or the observed calibrator can not be found in this file, no transfer function will be calculated.

The following command line can now be given at the shell prompt:

```
esorex --output-dir=Products_calib1 midi_fringe_all hs_calib1.sof
```

Two main products and two diagnostic files are created in `Products_calib1`:

MIDI_b1_hsc.pro.fits Primary product containing uncalibrated visibilities

MIDI_trf_prism.fits Transfer function

MIDI_b1_hsc.stat.fits File containing additional plotting information

MIDI_b1_hsc.waf.fits Waterfall image

The OI-Fits file `MIDI_b1_hsc.pro.fits` with the product category `REDUCED_DISPERSED` contains the reduced uncalibrated visibilities and the corresponding errors. Please note that the errors are given as relative errors, e.g. an error of 0.04 means an error of 4 per cent.
The transfer function is stored in file `MIDI_trf_prism.fits` with the product category `TRF_PRISM`. Moreover the transfer function file is automatically copied to `/tmp/` in order to be used in a subsequent science reduction with no transfer function in the SOF. This means that one has to manually remove the file from `/tmp/` if no calibration of the subsequent science observation is desired (please read carefully Section 10.5!).

The third step in the cascade is to reduce the science data and derive (if possible) calibrated visibilities by using the recipe *midi_fringe_all*.

The input SOF may be defined as follows:

File: *hs_science1_calibrated.sof*

```
MIDI.2007-06-30T07:25:24.000.fits     HIGH_SENS_SCIENCE
MIDI.2007-06-30T07:26:38.507.fits     HIGH_SENS_SCIENCE
MIDI.2007-06-30T07:27:53.013.fits     HIGH_SENS_SCIENCE
MIDI.2007-06-30T07:29:57.750.fits     PHOTOM_HS_SCIENCE
```

```
MIDI.2007-06-30T07:31:12.236.fits     PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:32:50.250.fits     PHOTOM_HS_SCIENCE
MIDI.2007-06-30T07:34:04.736.fits     PHOTOM_HS_SCIENCE
Products_mask/midi_profile_prism.fits KAPPA_HS_MASK_PRISM
Products_calib1/MIDI_trf_prism.fits   TRF_PRISM
```

In this SOF the transfer function file `MIDI_trf_prism.fits` is used to calibrate the visibilities. If no transfer function file is found, uncalibrated visibilities are derived.

The following command line (splitted into two lines for clarity) can now be given at the shell prompt:

```
esorex --output-dir=Products_science1_calibrated
                    midi_fringe_all hs_science1_calibrated.sof
```

One main products and three diagnostic files are created in `Products_science1_calibrated`:

MIDI_b1_hss.pro.fits Primary product containing calibrated visibilities

MIDI_b1_hss.stat.fits File containing additional plotting information

MIDI_b1_hss.waf.fits Waterfall image

MIDI_trf_prism.fits Transfer function file identical to the one given in the SOF

Please note that one should always check the header of the primary OI-Fits product `MIDI_b1_hss.pro.fits` in order to be sure that the derived visibilities are calibrated or not. This information is stored in the header keyword `HIERARCH ESO PRO TRF`. This keyword is set to `AVAILABLE` or `UNAVAILABLE`, accordingly. Moreover the transfer function file given as input to the recipe is also saved as a product in order to be able to associate to the calibrated visibility the corresponding transfer function file.

5.3 Example of SCI_PHOT data reduction using Esorex

In the following, a typical MIDI reduction procedure for data taken in SCI_PHOT mode is described.[4] It is assumed that the following data are available:

Photometric exposures:

```
MIDI.2007-06-30T05:40:08.897.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T05:40:50.886.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T05:41:32.876.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T05:43:21.639.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T05:44:03.629.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T05:44:45.618.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T05:18:09.529.fits  PHOTOM_SP_SCIENCE
```

[4]The procedure using *Gasgano* is conceptually identical.

```
MIDI.2007-06-30T05:18:51.518.fits  PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:19:33.508.fits  PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:22:08.163.fits  PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:22:50.153.fits  PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:23:32.142.fits  PHOTOM_SP_SCIENCE
```

Files withe interferometric and photometric information of a calibrator target:

```
MIDI.2007-06-30T05:35:06.015.fits  SCI_PHOT_CALIB
MIDI.2007-06-30T05:35:48.005.fits  SCI_PHOT_CALIB
MIDI.2007-06-30T05:36:29.994.fits  SCI_PHOT_CALIB
MIDI.2007-06-30T05:37:11.983.fits  SCI_PHOT_CALIB
MIDI.2007-06-30T05:37:53.973.fits  SCI_PHOT_CALIB
```

Files with interferometric and photometric information of a science target:

```
MIDI.2007-06-30T05:11:22.015.fits  SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:12:04.004.fits  SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:12:45.971.fits  SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:13:27.961.fits  SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:14:09.950.fits  SCI_PHOT_SCIENCE
```

File containing fundamental data of various calibrators. This file is delivered together with the pipeline:

```
calibrator_database_N.fits        CALIB_DATABASE_N
```

In the following, it is also assumed for simplicity that, in the *Esorex* configuration file, the flag *suppress-prefix* is set to `TRUE` and all raw files are located in the current directory. Moreover it is also assumed that the following directories are created in order to store the results of the different recipes:

```
Products_mask
Products_calib1
Products_science1_calibrated
```

As for the HIGH_SENS datareduction the first step is to generate a suitable mask/profile file by using the recipe *midi_profile*.

The input SOF may be defined as follows:

File: *sp_mask1.sof*

```
MIDI.2007-06-30T07:14:25.250.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T07:15:39.735.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T07:17:15.250.fits  PHOTOM_SP_CALIB
MIDI.2007-06-30T07:18:29.756.fits  PHOTOM_SP_CALIB
```

```
MIDI.2007-06-30T07:29:57.750.fits  PHOTOM_SP_SCIENCE
MIDI.2007-06-30T07:31:12.236.fits  PHOTOM_SP_SCIENCE
MIDI.2007-06-30T07:32:50.250.fits  PHOTOM_SP_SCIENCE
MIDI.2007-06-30T07:34:04.736.fits  PHOTOM_SP_SCIENCE
```

This is the list of the raw frames used to derive the 2D profile of the spectral dispersed signal. Following the optimal extraction technique the product is used in various recipes to extract the signal with an optimal SNR. Moreover the product of this recipe also masks scattered light from the delay line tunnel when extracting the signal.

The following command line can now be given at the shell prompt:

```
esorex --output-dir=Products_mask midi_profile sp_mask1.sof
```

One main product and several diagnostic files (mainly for check purposes) are created in `Products_mask`:

midi_profile_prism.fits Main recipe product containing the profile of the signal

image_AOPEN_DATA1_profile.fits profile in the DATA1 section with telescope A open
image_AOPEN_DATA1_signal.fits signal in the DATA1 section with telescope A open
image_AOPEN_DATA2_profile.fits profile in the DATA2 section with telescope A open
image_AOPEN_DATA2_signal.fits signal in the DATA2 section with telescope A open
image_AOPEN_DATA3_profile.fits profile in the DATA3 section with telescope A open
image_AOPEN_DATA3_signal.fits signal in the DATA3 section with telescope A open
image_AOPEN_DATA4_profile.fits profile in the DATA4 section with telescope A open
image_AOPEN_DATA4_signal.fits signal in the DATA4 section with telescope A open
image_BOPEN_DATA1_profile.fits profile in the DATA1 section with telescope B open
image_BOPEN_DATA1_signal.fits signal in the DATA1 section with telescope B open
image_BOPEN_DATA2_profile.fits profile in the DATA2 section with telescope B open
image_BOPEN_DATA2_signal.fits signal in the DATA2 section with telescope B open
image_BOPEN_DATA3_profile.fits profile in the DATA3 section with telescope B open
image_BOPEN_DATA3_signal.fits signal in the DATA3 section with telescope B open
image_BOPEN_DATA4_profile.fits profile in the DATA4 section with telescope B open
image_BOPEN_DATA4_signal.fits signal in the DATA4 section with telescope B open

As you can see from the SOF we combined the photometry files of the calibrator and the science target in order to improve the SNR of the product. This should be done only after running the recipe on the calib and science data separately and checking (by e.g. blinking the diagnostic files) that the profile is not moving between the different observations. Moreover, one can fine-tune the result by changing the default recipe option `--threshold`. This threshold is used by the recipe to discriminate between source and background pixels (in sigma units). A typical result of this recipe is shown in Fig. 11.1.2.

The second step in the data reduction cascade is to derive the kappa matrix by using the recipe *midi_kappamatrix*.

The input SOF may be defined as follows:

File: *sp_kappa1.sof*

```
MIDI.2007-06-30T05:40:08.897.fits      PHOTOM_SP_CALIB
MIDI.2007-06-30T05:40:50.886.fits      PHOTOM_SP_CALIB
MIDI.2007-06-30T05:41:32.876.fits      PHOTOM_SP_CALIB
MIDI.2007-06-30T05:43:21.639.fits      PHOTOM_SP_CALIB
MIDI.2007-06-30T05:44:03.629.fits      PHOTOM_SP_CALIB
MIDI.2007-06-30T05:44:45.618.fits      PHOTOM_SP_CALIB
Products_mask/midi_profile_prism.fits  KAPPA_SP_MASK_PRISM
```

The following command line can now be given at the shell prompt:

```
esorex --output-dir=Products_kappa1 midi_kappamatrix sp_kappa1.sof
```

One main product and many diagnostic files are created in `Products_kappa1`:

midi_kappamatrix_prism.fits Primary product containing the kappa matrix

midi_kappamatrix11.fits Kappamatrix-11

midi_kappamatrix11_filtered.fits Kappamatrix-11 Median filtered

midi_kappamatrix11_nomask.fits Kappamatrix-11 No mask used for signal extraction

midi_kappamatrix12.fits Kappamatrix-12

midi_kappamatrix12_filtered.fits Kappamatrix-12 Median filtered

midi_kappamatrix12_nomask.fits Kappamatrix-12 No mask used for signal extraction

midi_kappamatrix21.fits Kappamatrix-21

midi_kappamatrix21_filtered.fits Kappamatrix-21 Median filtered

midi_kappamatrix21_nomask.fits Kappamatrix-21 No mask used for signal extraction

midi_kappamatrix22.fits Kappamatrix-22

midi_kappamatrix22_filtered.fits Kappamatrix-22 Median filtered

midi_kappamatrix22_nomask.fits Kappamatrix-22 No mask used for signal extraction

The third step in the data reduction cascade is to reduce the calibrator observations using the recipe *midi_fringe_all*.

The input SOF may be defined as follows:

File: *sp_calib1.sof*

```
MIDI.2007-06-30T05:35:06.015.fits           SCI_PHOT_CALIB
MIDI.2007-06-30T05:35:48.005.fits           SCI_PHOT_CALIB
MIDI.2007-06-30T05:36:29.994.fits           SCI_PHOT_CALIB
MIDI.2007-06-30T05:37:11.983.fits           SCI_PHOT_CALIB
MIDI.2007-06-30T05:37:53.973.fits           SCI_PHOT_CALIB
Products_mask/midi_profile_prism.fits       KAPPA_SP_MASK_PRISM
Products_kappa1/midi_kappamatrix_prism.fits MIDI_KAPPAMATRIX_PRISM
calibrator_database_N.fits                    CALIB_DATABASE_N
```

In this SOF the `calibrator_database_N.fits` file is used to derive the transfer function to calibrate a science observation. If this file is not in the SOF or the observed calibrator can not be found in this file, no transfer function will be calculated.

The following command line can now be given at the shell prompt:

```
esorex --output-dir=Products_calib1 midi_fringe_all sp_calib1.sof
```

Two main products and two diagnostic files are created in `Products_calib1`:

MIDI_b1_spc.pro.fits Primary product containing uncalibrated visibilities

MIDI_trf_prism.fits Transfer function

MIDI_b1_spc.stat.fits File containing additional plotting information

MIDI_b1_spc.waf.fits Waterfall image

The OI-Fits file `MIDI_b1_spc.pro.fits` with the product category `REDUCED_DISPERSED_SCIPHOT` contains the reduced uncalibrated visibilities and the corresponding errors. Please note that the errors are given as relative errors, e.g. an error of 0.04 means an error of 4 per cent.
The transfer function is stored in file `MIDI_trf_prism.fits` with the product category `TRF_PRISM`. Moreover the transfer function file is automatically copied to `/tmp/` in order to be used in a subsequent science reduction with no transfer function in the SOF. This means that one has to manually remove the file from `/tmp/` if no calibration of the subsequent science observation is desired (please read carefully Section 10.5!).

The next step is to create the kappa matrix for the science observation by using the recipe *midi_kappamatrix*. Please note that in order to improve the SNR of the kappa matrix one can combine different observations. However this should be done only after comparing the kappamatrixes of the single reduction blocks and ensuring that there are no systematical differences by e.g. comparing the diagnostic files.

The input SOF may be defined as follows:

File: *sp_kappa2.sof*

```
MIDI.2007-06-30T05:18:09.529.fits      PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:18:51.518.fits      PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:19:33.508.fits      PHOTOM_SP_SCIENCE
```

```
MIDI.2007-06-30T05:22:08.163.fits      PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:22:50.153.fits      PHOTOM_SP_SCIENCE
MIDI.2007-06-30T05:23:32.142.fits      PHOTOM_SP_SCIENCE
Products_mask/midi_profile_prism.fits  KAPPA_SP_MASK_PRISM
```

The following command line can now be given at the shell prompt:

```
esorex --output-dir=Products_kappa2 midi_kappamatrix sp_kappa2.sof
```

One main product and many diagnostic files are created in `Products_kappa2` (see second step above)

The last step in the cascade is to reduce the science data and derive (if possible) calibrated visibilities by using the recipe *midi_fringe_all*.

The input SOF may be defined as follows:

File: *sp_science1_calibrated.sof*

```
MIDI.2007-06-30T05:11:22.015.fits           SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:12:04.004.fits           SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:12:45.971.fits           SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:13:27.961.fits           SCI_PHOT_SCIENCE
MIDI.2007-06-30T05:14:09.950.fits           SCI_PHOT_SCIENCE
Products_mask/midi_profile_prism.fits       KAPPA_SP_MASK_PRISM
Products_kappa2/midi_kappamatrix_prism.fits MIDI_KAPPAMATRIX_PRISM
Products_calib1/MIDI_trf_prism.fits         TRF_PRISM
```

In this SOF the transfer function file `MIDI_trf_prism.fits` is used to calibrate the visibilities. If no transfer function file is found, uncalibrated visibilities are derived.

The following command line (splitted into two lines for clarity) can now be given at the shell prompt:

```
esorex --output-dir=Products_science1_calibrated
                    midi_fringe_all sp_science1_calibrated.sof
```

One main products and three diagnostic files are created in `Products_science1_calibrated`:

MIDI_b1_sps.pro.fits Primary product containing calibrated visibilities

MIDI_b1_sps.stat.fits File containing additional plotting information

MIDI_b1_sps.waf.fits Waterfall image

MIDI_trf_prism.fits Transfer function file identical to the one given in the SOF

Please note that one should always check the header of the primary OI-Fits product `MIDI_b1_sps.pro.fits` in order to be sure that the derived visibilities are calibrated or not. This information is stored in the header keyword `HIERARCH ESO PRO TRF`. This keyword is set to `AVAILABLE` or `UNAVAILABLE`, accordingly. Moreover the transfer function file given as input to the recipe is also saved as a product in order to be able to associate to the calibrated visibility the corresponding transfer function file.

6 Known Problems

The current release of the MIDI pipeline has the following deficiencies:

- The errors of the kappa matrix are not derived by error propagation but a priori set to be 5 per cent.

7 Instrument Data Description

This Section mainly deals with the appropriate classification of data used by the pipeline. This is done automatically by *Gasgano* [9], but not by *Esorex* [8]. If the pipeline user has already classified their FITS files, there is no need to use the formal classification rules described in this Section.

MIDI data can be separated into *raw* frames and *product* frames. Raw frames are the unprocessed output of the MIDI instrument observations, while product frames are either the result of the MIDI pipeline processing (as reduced frames, master calibration frames, etc.), or are outsourced (as the calibrator database, etc.).

Any *raw* or *calib* frame can be classified on the basis of a set of keywords read from its header. Data classification is typically carried out by the DO or by *Gasgano*, which apply the same set of classification rules. The association of a raw frame with calibration data (*e.g.*, of a science frame with a kappamatrix frame) can be obtained by matching the values of appropriate sets of header keywords.

Each kind of *raw* frame is typically associated to a single MIDI pipeline recipe, i.e., the recipe assigned to the reduction of that specific frame type. In the pipeline environment this recipe would be launched automatically.

A *calib* frame may be input to more than one MIDI pipeline recipe, but it may be created by just one pipeline recipe. In the automatic pipeline environment a *calib* data frame alone would not trigger the launch of any recipe.

A *product* frame may be input to more than one MIDI pipeline recipe, but it may be created by just one pipeline recipe. In the automatic pipeline environment a product data frame alone wouldn't trigger the launch of any recipe.

The format for the *product* frame is fully described in [6]. This document provides a reference source for the specification of the FITS files that are used and produced by the VLTI software. In order to avoid inconsistency as a result of duplication, the reader is advised to refer to this document.

In the following all raw MIDI data frames are listed, together with the keywords used for their classification and correct association. The indicated *DO category* is a label assigned by the online pipeline system to any data type after it has been classified, which is then used to identify the frames listed in the *Set of Frames* (see Section 5.1.2, page 20).

The *calib* frames produced by the pipeline are listed in the description of the individual recipes producing them. The *calib* frames which are used by the pipelines, indicated also as static calibration data, are described in section 8, page 36.

- **High Sens calibrator interferometry data:**

  DO category: `HIGH_SENS_CALIB`
  Processed by: `midi_fringe_all`

  Classification keywords:
  ```
  DPR CATG = CALIB
  DPR TECH = INTERFEROMETRY
  DPR TYPE = TRACK,OBJECT,DISPERSED
  TPL NAME = MIDI STARINTF OBS FRINGE
  INS OPT1 ID = HIGH_SENS
  ```

- **High Sens calibrator photometry data:**
  DO category: `PHOTOM_HS_CALIB`
  Processed by: `midi_fringe_all, midi_profile`

  Classification keywords:
  ```
  DPR CATG = CALIB
  DPR TECH = IMAGE,WINDOW,CHOPNOD
  DPR TYPE = PHOTOMETRY,OBJECT
  TPL NAME = MIDI STARINTF OBS FRINGE
  INS OPT1 ID = HIGH_SENS
  ```

- **High Sens science interferometry data:**
  DO category: `HIGH_SENS_SCIENCE`
  Processed by: `midi_fringe_all`

  Classification keywords:
  ```
  DPR CATG = SCIENCE
  DPR TECH = INTERFEROMETRY
  DPR TYPE = TRACK,OBJECT,DISPERSED
  TPL NAME = MIDI STARINTF OBS FRINGE
  INS OPT1 ID = HIGH_SENS
  ```

- **High Sens science photometry data:**
  DO category: `PHOTOM_HS_SCIENCE`
  Processed by: `midi_fringe_all, midi_profile`

  Classification keywords:
  ```
  DPR CATG = SCIENCE
  DPR TECH = IMAGE,WINDOW,CHOPNOD
  DPR TYPE = PHOTOMETRY,OBJECT
  TPL NAME = MIDI STARINTF OBS FRINGE
  INS OPT1 ID = HIGH_SENS
  ```

- **Sci Phot calibrator interferometry data:**
  DO category: `SCI_PHOT_CALIB`
  Processed by: `midi_fringe_all`

  Classification keywords:
  ```
  DPR CATG = CALIB
  DPR TECH = INTERFEROMETRY
  DPR TYPE = TRACK,OBJECT,DISPERSED,SCIPHOT
  TPL NAME = MIDI STARINTF OBS FRINGE
  INS OPT1 ID = SCI_PHOT
  ```

- **Sci Phot calibrator photometry data:**
  DO category: `PHOTOM_SP_CALIB`
  Processed by: `midi_fringe_all, midi_kappamatrix, midi_profile`

Classification keywords:
```
DPR CATG = CALIB
DPR TECH = IMAGE,WINDOW,CHOPNOD
DPR TYPE = PHOTOMETRY,OBJECT,SCIPHOT
TPL NAME = MIDI STARINTF OBS FRINGE
INS OPT1 ID = SCI_PHOT
```

- **Sci Phot science interferometry data:**
  DO category: `SCI_PHOT_SCIENCE`
  Processed by: `midi_fringe_all`

  Classification keywords:
```
DPR CATG = SCIENCE
DPR TECH = INTERFEROMETRY
DPR TYPE = TRACK,OBJECT,DISPERSED,SCIPHOT
TPL NAME = MIDI STARINTF OBS FRINGE
INS OPT1 ID = SCI_PHOT
```

- **Sci Phot science photometry data:**
  DO category: `PHOTOM_SP_SCIENCE`
  Processed by: `midi_fringe_all, midi_kappamatrix, midi_profile`

  Classification keywords:
```
DPR CATG = SCIENCE
DPR TECH = IMAGE,WINDOW,CHOPNOD
DPR TYPE = PHOTOMETRY,OBJECT,SCIPHOT
TPL NAME = MIDI STARINTF OBS FRINGE
INS OPT1 ID = SCI_PHOT
```

- **Acquisition:**
  DO category: `ACQ`
  Processed by: `midi_acq`

  Classification keywords:
```
DPR CATG = SCIENCE |
DPR CATG = CALIB
DPR TECH = IMAGE,WINDOW
DPR TYPE = COARSE,OBJECT
```

- **Detector linearity:**
  DO category: `DETLIN`
  Processed by: `midi_detlin`

  Classification keywords:
```
DPR CATG = CALIB
DPR TECH = SPECTRUM
DPR TYPE = FLAT
```

- **Detector readout noise:**
  DO category: `DETRON`
  Processed by: `midi_detron`

  Classification keywords:
  `DPR CATG = CALIB`
  `DPR TECH = IMAGE`
  `DPR TYPE = BIAS`

- **Dispersed transmission:**
  DO category: `DSPTRN`
  Processed by: `midi_dsptrn`

  Classification keywords:
  `DPR CATG = CALIB`
  `DPR TECH = SPECTRUM`
  `DPR TYPE = WAVE`

- **Reference pixel:**
  DO category: `REFPIX`
  Processed by: `midi_refpix`

  Classification keywords:
  `DPR CATG = CALIB`
  `DPR TECH = IMAGE`
  `DPR TYPE = FMTCHCK`

- **Wavelength calibration:**
  DO category: `WAVECAL`
  Processed by: `midi_wavecal`

  Classification keywords:
  `DPR CATG = CALIB`
  `DPR TECH = SPECTRUM`
  `DPR TYPE = WAVE,SPECTEMPL`

- **Internal OPD accuracy:**
  DO category: `INTERNAL_OPD`
  Processed by: `midi_intopd`

  Classification keywords:
  `DPR CATG = CALIB`
  `DPR TECH = INTERFEROMETRY`
  `DPR TYPE = FRINGE,LAMP`

8 Static Calibration Data

In this section the MIDI static calibration files are listed. The indicated *DO category* (for fits files), written in the FITS header keyword PRO.CATG, is a label assigned to any data type after it has been classified. This label is then used to identify the frames listed in the *set-of-frames* (see Section 5.1.2, page 20).

8.1 Calibrator Database

DO category: `CALIB_DATABASE_N`

This fits table contains a rather comprehensive list of targets and their associated parameters such as RA, DEC, diameter etc. The file is used to compute the *Transfer Functions* if a HIGH_SENS or a SCI_PHOT CALIB target is observed. The recipe uses the astrometric position of the target to extract (if possible) all relevant informations from the file in order to compute the *expected visibilities* of the latter. From the expected visibilities and the observed uncalibrated visibilities it derives the *Transfer Function*. If the subsequent observation is a relevant SCIENCE target, the *Transfer Function* is used to compute the *Calibrated Visibilities*. The filename of the currently available calibrator database is *calibrator_database_N.fits*.

8.2 Wavelength Calibration Database

There are four *Wavelength Calibration* files:

waveCalib_GRISM_HIGH_SENS.dat
waveCalib_PRISM_HIGH_SENS.dat
waveCalib_GRISM_SCI_PHOT.dat
waveCalib_GRISM_SCI_PHOT.dat

These ASCII files contain, in a number of columns, the *Calibrated Wavelengths* for all the *Detector Regions*. The number of columns correspond to the number of the *Detector Regions*. For HIGH_SENS data there are two and for SCI_PHOT data there are four columns in each file. The MIDI pipeline automatically uses the relevant file when computing the *Visibilities*

9 Data Reduction

This section provides an overview of the MIDI Data Reduction. It outlines the main objectives of the program, gives an overview of the input data and their interdependence with various recipes.

9.1 Data reduction overview

The current MIDI pipeline offers the following recipes:

midi_raw_to_fitsimage, for converting midi raw files into data cubes. Moreover, the recipe calculates up to 28 (depending on the instrument mode) diagnostic files.

midi_profile, to derives the 2D profile/mask of the spectral dispersed signal in the SCI_PHOT and HIGH_SENS mode. The product is used for other recipes to perform an optimally weighted signal extraction.

midi_kappamatrix, to derive the kappamatrix following the definition in [10], needed to perform the flux compensation in the SCI_PHOT mode.

midi_fringe_all, to compute the Visibilities. Depending on the input files and reduction cascade, the recipe calculates uncalibrated Visibilities, Transfer Functions and calibrated Visibilities for HIGH_SENS and SCI_PHOT data.

midi_acq, to evaluate the image quality, i.e. the position, size and the flux intensity of the target.

midi_detron, to evaluate the detectors readout noise.

midi_detlin, to evaluate the detector linearity.

midi_refpix, evaluates the alignment (reference positions) of the VLTI beams.

midi_dsptrn, to evaluate the transmission quality of the dispersive elements.

midi_wavecal, to produce a wavelength calibration. database

midi_intopd, to evaluate the stability of the internal OPD by using the algorithm described in [11].

Detailed description of these recipes are given in section 10 and 11.

9.2 Reduction Cascade

The possible data reduction paths which can be followed using the data reduction recipes are shown in image Fig. 9.2.1. Moreover a typical reduction procedure of HIGH_SENS ans SCI_PHOT data is given in Section 5.

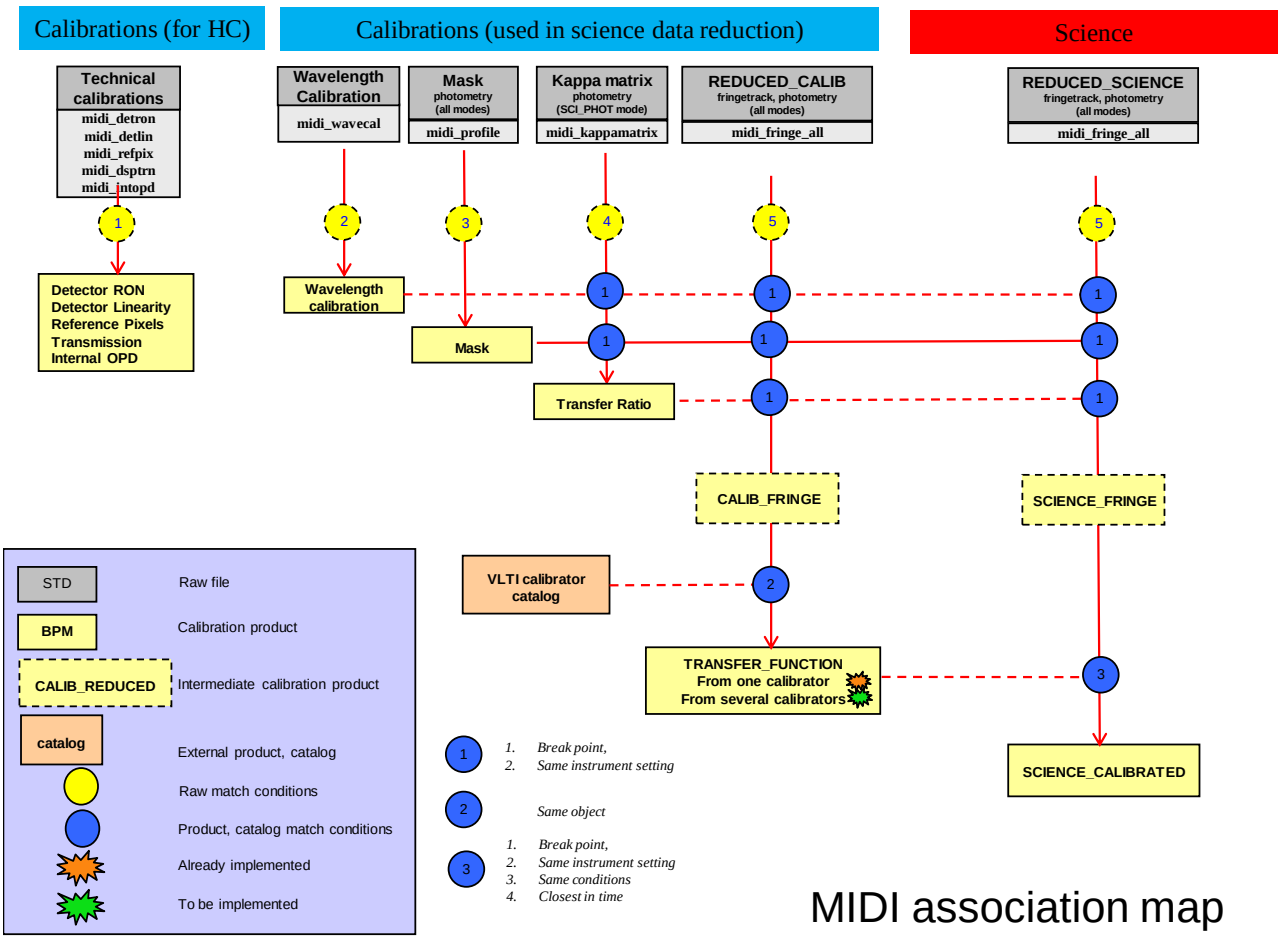

Figure 9.2.1: *Data reduction flow using the MIDI recipes*

10 Pipeline Recipes Interfaces

In this Section a detailed description of the MIDI pipeline recipes interfaces is given, with a complete specification of the recipes usage, their input, output, and configuration parameters. For a overview of the available pipeline recipes, please see Section 9, page 37.

10.1 midi_profile

This recipe derives the 2D profile of the spectral dispersed signal in the SCI_PHOT (SP) and HIGH_SENS (HS) mode. Following the optimal extraction technique the product is used in various recipes to extract the signal with an optimal SNR. Moreover the product of this recipe also masks scattered light from the delay line tunnel when extracting the signal.

10.1.1 Input files

At least two single photometry files -one for telescope A open (header keyword INS.SHUT.ID=AOPEN), and one for telescope B open (INS.SHUT.ID=BOPEN)- must be included in the SOF. Therefore they must belong to one of the following DO categories:

- **PHOTOM_SP_CALIB:** photometry frame in the SCI_PHOT mode
- **PHOTOM_SP_SCIENCE:** photometry frame in the SCI_PHOT mode
- **PHOTOM_HS_CALIB:** photometry frame in the HIGH_SENS mode
- **PHOTOM_HS_SCIENCE:** photometry frame in the HIGH_SENS mode

10.1.2 Output files

Depending on the input frames (HIGH_SENS, SCI_PHOT, GRISM, PRISM) the main product of the recipe is a fits file with one of the following DO categories (check header keyword PRO.CATG):

- **KAPPA_HS_MASK_PRISM:** Primary output for HS PRISM mode. The file contains the spatial profile of the source and should be used to optimally extract the signal in various other recipes.
- **KAPPA_SP_MASK_PRISM:** Primary output for SP PRISM mode. The file contains the spatial profile of the source and should be used to optimally extract the signal in various other recipes.
- **KAPPA_HS_MASK_GRISM:** Primary output for HS GRISM mode. The file contains the spatial profile of the source and should be used to optimally extract the signal in various other recipes.
- **KAPPA_SP_MASK_GRISM:** Primary output for SP GRISM mode. The file contains the spatial profile of the source and should be used to optimally extract the signal in various other recipes.

Additional to the above mentioned main products, the recipe also creates diagnostic images. Depending on the observation mode (HS or SP) and on the used grism (GRISM or PRISM) the following secondary products are created:

File name:	DO category:
image_AOPEN_DATA[x]_profile.fits	KAPPA_[mode]_MASK_[grism]_PROFILE
image_AOPEN_DATA[x]_signal.fits	KAPPA_[mode]_MASK_[grism]_SIGNAL
image_BOPEN_DATA[x]_profile.fits	KAPPA_[mode]_MASK_[grism]_PROFILE
image_BOPEN_DATA[x]_signal.fits	KAPPA_[mode]_MASK_[grism]_SIGNAL

with [grism] ⇒ GRISM/PRISM, [x] ⇒ 1/2/3/4, and [mode] ⇒ HS/SP
If the filename or the DO category of the image is ending in -signal/SIGNAL-, the spectroscopic flux of the source is shown in the fits-file. If the file-name or the DO category is ending in -profile/PROFILE-, the profile normalized to unity perpendicular to the dispersion direction is shown. The latter is a copy of the profile stored in the main recipe product.

Typical examples of the recipe output can be found in Fig. 11.1.1 and Fig. 11.1.2.

10.1.3 Recipe parameters

The configuration parameters of a recipe determine the way the recipe will process the input frames. A single option is provided by this recipe:

`--threshold`: The threshold in sigma units to discriminate between source/profile and background pixels.

10.1.4 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.1.

10.2 midi_kappamatrix

This recipe derives the kappamatrix in the SCI_PHOT mode as defined in [10]. It processes a set of AOPEN and BOPEN photometry files as well as a spatial profile of the spectrum (see section 10.1).

10.2.1 Input files

At least two single SCI_PHOT photometry files -one for telescope A open (AOPEN), and one for telescope B open (BOPEN)- together with a profile frame must be included in the SOF. Moreover, they must belong to the DO categories listed in the following table:

- **PHOTOM_SP_CALIB:** SCI_PHOT photometry frames
- **PHOTOM_SP_SCIENCE:** SCI_PHOT photometry frames
- **KAPPA_SP_MASK_PRISM:** Source profile frame
- **KAPPA_SP_MASK_GRISM:** Source profile frame

10.2.2 Output files

Depending on the input frames (GRISM or PRISM) the main product of the recipe is is a fits file with one of the following DO categories (check header keyword PRO.CATG):

- **MIDI_KAPPAMATRIX_GRISM:** File containing the kappa matrix for the GRISM mode.
- **MIDI_KAPPAMATRIX_PRISM:** File containing the kappa matrix for the PRISM mode.

The spectrum is extracted using the Optimal Extraction algorithm [12] adapted to NIR data, i.e. background dominated images (see also section 11.2). Please note that the errors of the kappa matrix are not calculated but set a priori to 5 per cent.
Additional to the above mentioned main products, the recipe also creates diagnostic images. The following diagnostic products are created:

File name:	DO category:
midi_kappamatrix[x].fits	MIDI_KAPPAMATRIX[x]
midi_kappamatrix[x]_filtered.fits	MIDI_KAPPAMATRIX[x]_FILTERED
midi_kappamatrix[x]_nomask.fits	MIDI_KAPPAMATRIX[x]_NOMASK

with [x] $\Rightarrow$ 11/12/21/22
The image midi_kappamatrix[x].fits (MIDI_KAPPAMATRIX[x]) is a copy of the kappamatrix stored in the main product.

If the filename or the DO category of the image is ending in -filtered/FILTERED-, the kappamatrix is smoothed by a median filter. By using the recipe option `midi.midi_kappamatrix.medianwindow` (gasgano) or `--medianwindow` (esorex) the window size of the median filter can be set.
If the filename or the DO category of the image is ending in -nomask/NOMASK-, the kappamatrix is extracted without using the profile frame in the SOF, i.e. extracting the spectral dispersed signal without applying an optimal extraction algorithm

A typical kappa matrix (recipe output) can be found in Fig. 11.2.1 and Fig. 11.2.2.

10.2.3 Recipe parameters

Although a single option is provided by this recipe, this option has **no** impact to the primary fits-table product, but only to the diagnostic file with the DO category MIDI_KAPPAMATRIX[x]_FILTERED.

`--medianwindow`: The window size of the median filter.

10.2.4 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.2.

10.3 midi_raw_to_fitsimage

The main purpose of this recipe is to convert the imaging sections of the Midi raw files into fits-cubes or fits-images.

10.3.1 Input files

This recipe is able to process all midi raw fitsfiles with an DATA1/2/3/4 column in the IMAGING_DATA extension, i.e. most of the midi raw files. Therefore no special classification tag needs to be given to the SOF. Moreover, the SOF should include only one midi raw file.

10.3.2 Output files

Depending on the input frames (chopped or not chopped) the main products of the recipe are fits files with the following DO categories (check header keyword PRO.CATG):

File name	DO category	Type
• DATA[x]_cube.fits	MIDI_CUBE	cube
• DATA[x]_sky_cube.fits	MIDI_CUBE_SKY	cube
• DATA[x]_target_cube.fits	MIDI_CUBE_TARGET	cube
• DATA[x]_raw_timecollapsed_cube.fits	MIDI_RAWTIMECOLLAPSED	image
• DATA[x]_result_cube.fits	MIDI_CUBE_TARGET_MINUS_SKY	cube
• DATA[x]_result_timecollapsed_cube.fits	MIDI_TIMECOLLAPSED	image
• DATA[x]_result_timecollapsed_ycollapsed_cube.fits	MIDI_TIMECOLLAPSED_YCOLLAPSED	image

with [x] $\Rightarrow$ 1/2/3/4
If the raw file is not choped, only the files with the DO category MIDI_CUBE and MIDI_RAWTIMECOLLAPSED are created. If the raw file is chopped, i.e. contains sequences of *target* and *sky* exposures, more products are derived.

The differend DO categories read:

- **MIDI_CUBE :** The raw file has been converted into a fits cube. All frames are saved.
- **MIDI_CUBE_SKY :** The raw file has been converted into a fits cube. All frames marked as *sky* are saved.
- **MIDI_CUBE_TARGET :** The raw file has been converted into a fits cube. All frames marked as *target* are saved.
- **MIDI_RAWTIMECOLLAPSED :** The cube with the DO category MIDI_CUBE is averaged in the time domain and saved as an image file.
- **MIDI_CUBE_TARGET_MINUS_SKY :** The cube with the DO category MIDI_CUBE_SKY has been subtracted from the cube with the DO category MIDI_CUBE_TARGET frame by frame and saved in a cube.

- **MIDI_TIMECOLLAPSED :** The cube with the DO category MIDI_CUBE_TARGET_MINUS_SKY is averaged in the time domain and saved as an image file.
- **MIDI_TIMECOLLAPSED_YCOLLAPSED:** The cube with the DO category MIDI_TIMECOLLAPSED is summed up in the y direction and saved as a image file.

10.3.3 Recipe parameters

A single option is provided by this recipe:

`--outfile`: The output filename post-fix.

10.4 midi_intopd

This recipe calculates the internal OPD stability of MIDI based on the group delay [13]. It analizes the fringes in the frequency domain by the usage of the Scargle-Lomb algorithm [11].

10.4.1 Input files

At least one interferometric (ABOPEN) file must be included in the SOF. Moreover, the frame(s) must belong to the following DO category:

- **INTERNAL_OPD:** interferometry frame with high SNR lamp exposures.

10.4.2 Output files

- **MIDI_INTOPD:** This file contains the time-stamp, the path difference of the OPD, as well as the power of the Lomb-Scargel analysis.

Additional to the above mentioned main product, the following secondary products are created:

File name:	DO category:	Type
opdmask.fits	MIDI_MASK_INTOPD	table
opdmask_DATA1.fits	MIDI_MASK_DATA1	image
opdmask_DATA2.fits	MIDI_MASK_DATA2	image

with MIDI_MASK_INTOPD (fits table) being the mask file used to extract the signal of DATA1 and DATA2 , MIDI_MASK_DATA1 (fits image) being the mask file used to extract the signal of DATA1, and MIDI_MASK_DATA2 (fits image) being the mask file used to extract the signal of DATA2.

A typical example of the recipe output can be found in Fig. 11.3.2

10.4.3 Recipe parameters

The following options are supported by this recipe:

`--llx`: Lower left x-value of the window where the signal resides.

`--lly`: Lower left y-value of the window where the signal resides.

`--urx`: Upper right x-value of the window where the signal resides.

`--ury`: Upper right y-value of the window where the signal resides.

`--fmin`: Minimum frequency of the Lomb evaluation window.

`--fmax`: Maximum frequency of the Lomb evaluation window.

`--sampling`: Sampling rate inside the Lomb evaluation window.

Please note that the minimum and maximum frequency of the lomb evaluation window should be set to reasonable values. If, for example, a mean path difference of 140 micron is expected `--fmin` and `--fmax` should be set to about 100 and 200.

10.4.4 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.3.

10.5 midi_fringe_all

This recipe is able to compute *Uncalibrated Visibilities*, *Transfer Functions* and *Calibrated Visibilities* for HIGH_SENS and SCI_PHOT data.

If a calibrator listed in the calibrator database (see section 8.1) is observed, the *Uncalibrated Visibilities* together with the *Transfer Function* are derived. In the pipeline installed at Paranal, the *Transfer Function* will subsequently be automatically used for a related SCIENCE observation (immediately following the calibrator observation!) to compute the *Calibrated Visibilities* of a SCIENCE target.

10.5.1 Input for HIGH_SENS data

- **HIGH_SENS_CALIB or HIGH_SENS_SCIENCE** *required* set of splitted raw, unprocessed interferometry data taken in HIGH_SENS mode.
- **PHOTOM_HS_CALIB or PHOTOM_HS_SCIENCE** *required* set of splitted raw, unprocessed photometry data taken in HIGH_SENS mode.
- **KAPPA_HS_MASK_GRISM and/or KAPPA_HS_MASK_PRISM** *required* file(s) containing the normalized 2D profile of the spectral dispersed signal. This file(s) is used to extract the signal with a high SNR following the optimal extraction technique. Please note, that if the dispersive element of the observations is GRISM (PRISM) the file with the DO classification KAPPA_HS_MASK_GRISM (KAPPA_HS_MASK_PRISM) is required. The pipeline will automatically select the suitable profile if both are given in the SOF.
- **TRF_GRISM and/or TRF_PRISM** *optional* file(s) containing the transfere function to be applied to calibrate HIGH_SENS_SCIENCE data. Please note, that if the dispersive element of the observations is GRISM (PRISM) the file with the DO classification TRF_GRISM (TRF_PRISM) is required. The pipeline will automatically select the suitable profile if both are given in the SOF.
- **CALIB_DATABASE_N** *optional* fits-table containing a list of calibrator targets and their associated parameters. The list is used by the pipeline to calculate the expected visibility of a calibrator and from the latter together with the measured visibility the *Transfer Function*. If the fits-table is not provided, no *Transfer Function* will be calculated.

10.5.2 Input for SCI_PHOT data

- **SCI_PHOT_CALIB or SCI_PHOT_SCIENCE** *required* set of splitted raw, unprocessed interferometry data taken in SCI_PHOT mode.
- **MIDI_KAPPAMATRIX_GRISM and/or MIDI_KAPPAMATRIX_PRISM** , *optional* fits-table containing the kappa matrix used to perform the *Optical Compensation*. If this information is not provided in the SOF, no optical compensation will be applied to the product visibilities.
- **KAPPA_SP_MASK_GRISM and/or KAPPA_SP_MASK_PRISM** *required* file(s) containing the normalized 2D profile of the spectral dispersed signal. This file(s) is used to extract the signal with a high SNR following the optimal extraction technique. Please note, that if the dispersive element of the observations is GRISM (PRISM) the file with the DO classification KAPPA_SP_MASK_GRISM (KAPPA_SP_MASK_PRISM) is required. The pipeline will automatically select the siutable profile if both are given in the SOF.

- **TRF_GRISM and/or TRF_PRISM** *optional* file(s) containing the transfer function to be applied to calibrate HIGH_SENS_SCIENCE data. Please note, that if the dispersive element of the observations is GRISM (PRISM) the file with the DO classification TRF_GRISM (TRF_PRISM) is required. The pipeline will automatically select the siutable profile if both are given in the SOF.

- **CALIB_DATABASE_N** *optional* fits-table containing a list of calibrator targets and their associated parameters. The list is used by the pipeline to calculate the expected visibility of a calibrator and from the latter together with the measured visibility the *Transfer Function*. If the fits-table is not provided, no *Transfer Function* will be calculated.

Please note:

1. If the midi_fringe_all recipe processes a calibrator and finds the latter in the calibrator database, it derives and saves a transfer function to a fits file with the following DO category: TRF_GRISM or TRF_PRISM.

2. This fits file can then be included in the SOF of a science observation in order to calculate calibrated (squared) visibilities.

3. The transfer function file is also (automatically) stored under /tmp and used for the following science observation. Once used, it is automatically deleted from /tmp, i.e. the transfer function file is only valid for one(!) science observation taken after the calibrator observation. This is important for the on-line pipeline installed at Paranal.

4. If the transfer function file is already present in the SOf of a science observation, no check for a file in /tmp will be done, i.e. the one in the SOF will be used to derive calibrated visibilities.

5. If the science observation is calibrated with a transfer function, the transfer function is also saved as a new product with the DO category TRF_GRISM or TRF_PRISM. This ensures that the associated transfer function used to calibrate a science observation can be easily identified later on.

10.5.3 Output files

- **REDUCED_DISPERSED:** Main MIDI products with the calculated (squared) visibilities. The fits file is compatible with the latest OI_FITS standard.

- **REDUCED_STAT_SP or REDUCED_STAT_HS:** This file is a copy of the main MIDI product (REDUCED_DISPERS with 3 additional extensions. This three extensions contain adittional informations and are added mostly for plotting reasons.

- **WATERFALL** This file visualizes the midi fringes per batch (x-coordinate of the piezo modulating the OPD) and per scan (y-coordinate), i.e. it represents the modulated fringes. Please note that this file is only used for data visualization and diagnostic.

- **TRF_GRISM or TRF_PRISM:** *optional* fits-table containing the transfer function. *Case A*): If a calibrator is observed and found in the calibrator database (CALIB_DATABASE_N) the transfer function is calculated and saved as a pipeline product. Moreover, this file is also automatically stored under

/tmp/MIDI_trf.fits. This is important for the on-line pipeline at Paranal to calibrate the subsequent science observation. *Case B*): If a science target is observed and a transfere function is provided to calculate calibrated visibilities, the transfer function is also saved as a new product. This ensures that the associated transfer function used to calibrate a science observation can be easily identified later on.

- **CORRELATED_FLUX:** The product contains a table with the wavelength, the correlated flux, the error of the correlated flux and a flag (flag equals to zero if the corresponding channel could be properly reduced). Please note that this new product is still under evaluation.

10.5.4 Recipe parameters

The configuration parameters of a recipe determine the way the recipe will process the input frames. A single option is provided by this recipe:

`--checkSof`: Do homogeneity checks on the SOF. If deactivated (checkSof=FALSE) the pipeline will not check if all rawfiles in the SOF have the same TPL START or a continuous OCS TPL FILENO header keyword. This allows expert users to combine and process files coming from different templates. USE WITH CAUTION!!

10.5.5 Quality control parameters

Currently more than 100 QC parameters, used by PSO and DFO, are calculated by this recipe. A list of the calculated Quality Control parameters together with as a short description can be found in Appendix C.4.

10.6 midi_acq

The purpose of this recipe is to assess the position, size and the flux intensity of the target.

10.6.1 Input files

- **ACQ** *required* file with several extensions. The important extension is the IMAGING_DATA which, amongst others, consists of the two detector regions DATA1 and DATA2.

10.6.2 Output files

- **IMAGE_QUALITY** At the moment two files are created with this `PRO CATG` header keyword: `MIDI_b1_acq_DATA1.pro.fits` and `MIDI_b1_acq_DATA2.pro.fits`. The averaged images of the detector regions 1 and 2 are shown in Fig. 10.6.1.
- **MIDI_ACQ_FOV_DATA1** Field of view of beam 1 (see also new QC parameter)
- **MIDI_ACQ_FOV_DATA2** Field of view of beam 2 (see also new QC parameter)

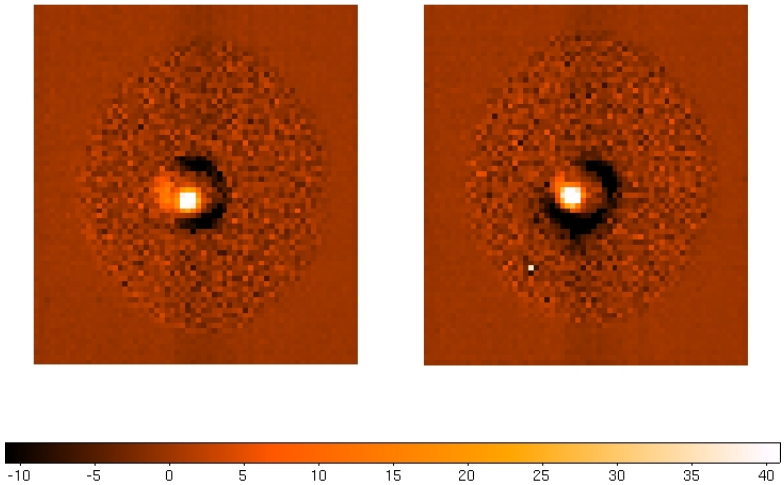

Figure 10.6.1: *Averaged image of the detector region DATA1 (left panel) and DATA2 (right panel)*

10.6.3 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.5. Please note, that some QC parameter related to the gaussian fitting of the field of view are derived only for AT observations. For UT observations these values are not important and thus the algorithm was designed to optimally work only for AT observations.

10.7 midi_wavecal

The purpose of this recipe is to facilitate calibration of the detector channels.

10.7.1 Input files

- **WAVECAL:** *6 required* files. Each FITS file has several extensions. The important extension is the IMAGING_DATA which amongst others consists of the image data relating to the detector regions (two regions for HIGHSENS and four regions for SCIPHOT). The six data files are taken with the following filters:
  - ArIII
  - NeII
  - SIV
  - Policarbonate Foil
  - OPEN
  - CLOSED

An example of the differend images is shown in Fig. 10.7.1.

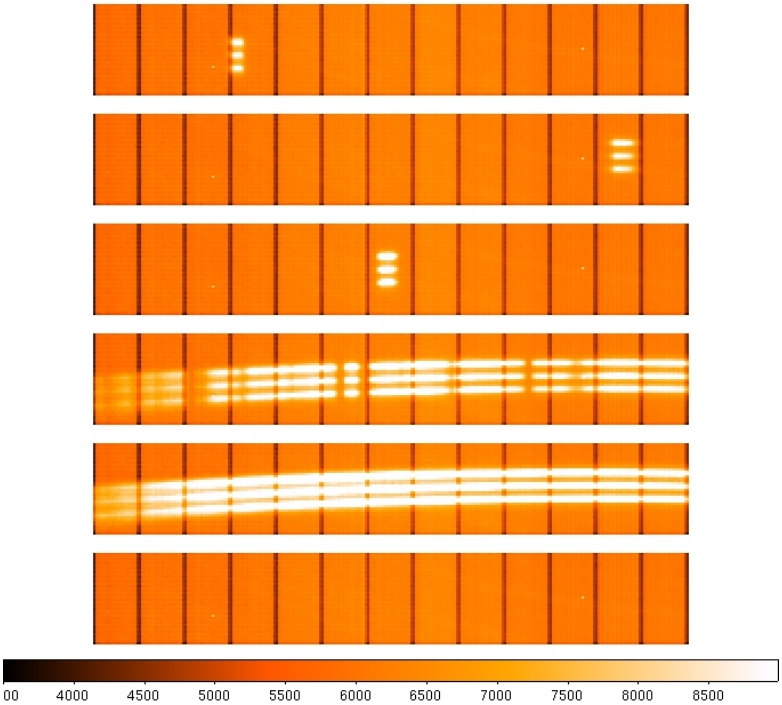

Figure 10.7.1: *Averaged images of the detector region DATA1 for the differend filters. From top to bottom: ArIII, NeII, SIV, Policarbonate Foil, OPEN, CLOSED*

10.7.2 Output files

• **REDUCED_WAVECAL:** Contains the wavelength calibration in the header.

10.7.3 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.10.

10.8 midi_detlin

The purpose of this technical recipe is to evaluate the relationship between the detector counts and the number of photons.

10.8.1 Input files

- **DETLIN** 12 *required* files with different exposure times. The important extension is the IMAGING_DATA which, amongst others, consists of the two detector regions DATA1 and DATA2.

An example of a typical images is shown in Fig. 10.8.1.

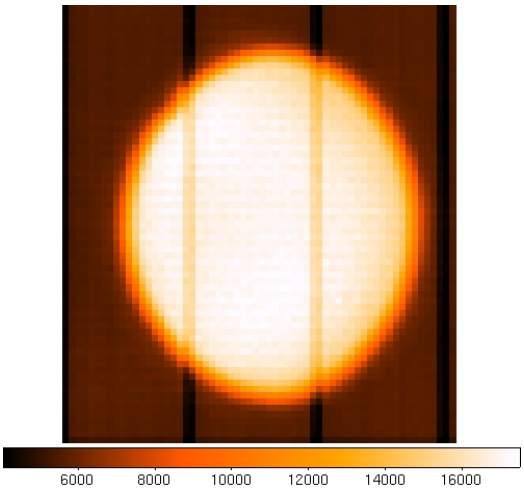

Figure 10.8.1: *Averaged images of the detector region DATA1 used as input for the midi_detlin recipe*

10.8.2 Output files

- **REDUCED_DETLIN:** The header of this fits file contains the coefficients of the polynomial describing the detector linearity and the image provides the graph of the detector linearity.

10.8.3 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.6.

10.9 midi_detron

The purpose of this technical recipe is to evaluate the read-out noise of the detector.

10.9.1 Input files

- **DETRON** 5 *required* BIAS files with several extensions. The important extension is the IMAGING_DATA which, consists of the the full detector.

10.9.2 Output files

- **REDUCED_DETRON:** contains in the header the median of the noise in the detector region. Moreover, the image part provides the detector readout noise (see Fig. 10.9.1).

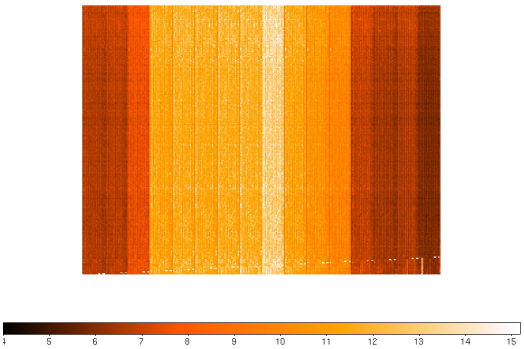

Figure 10.9.1: *Averaged detector-readout-noise image of the detector*

10.9.3 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.7.

10.10 midi_dsptrn

The purpose of this technical template is to measure the transmission characteristics of the dispersive elements (Prism and Grism) in order to monitor the evolution of their coatings.

10.10.1 Input files

- **DSPTRN** *required* set of 6 raw data (black body source). Files 1, 2, and 3 are take in imaging mode with the filter ArIII, NeII, and SIV, respectively. No dispersive element is used for file 1, 2 and 3. File 4, 5, and 6 are taken in spectroscopy mode with the filter ArIII, NeII, and SIV, respectively. A dispersive element (grism or prism) is used for these files.

An example of the images is shown in Fig. 10.10.1 for PRISM data.

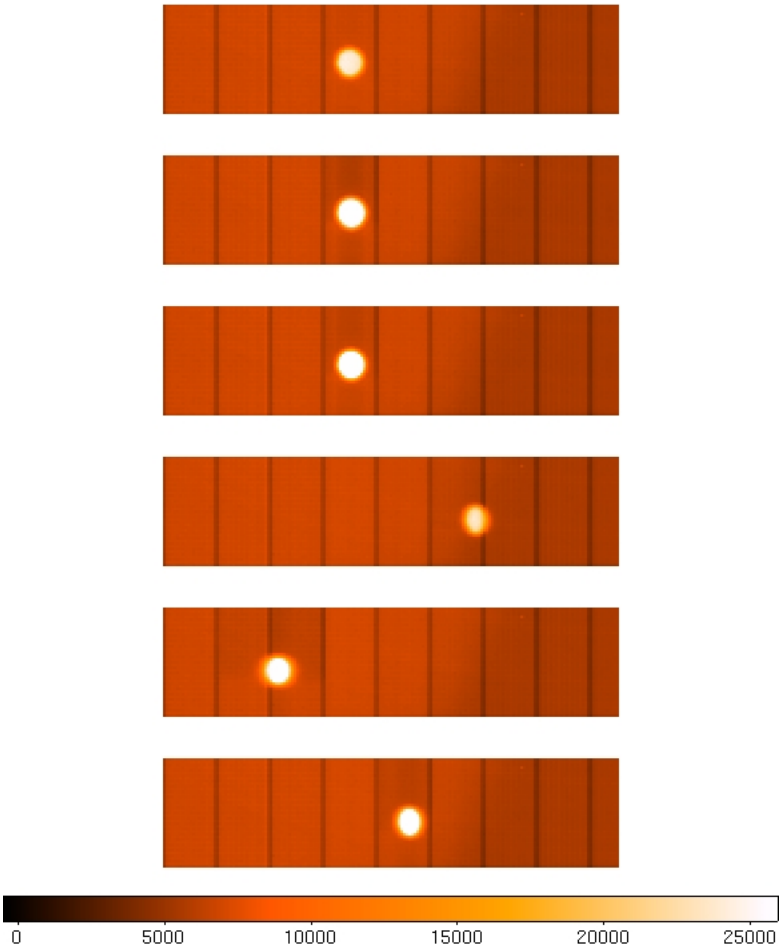

Figure 10.10.1: *Averaged images of the detector region DATA1 for the differend filters. From top to bottom: ArIII, NeII, and SIV without dispersive element; ArIII, NeII, and SIV with dispersive element (PRISM)*

10.10.2 Output files

- **REDUCED_DSPTRN:** Contains the dispersed transmission of the dispersive element (Grism or Prism) in the three filters (ArIII, NeII, SIV).

10.10.3 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.8.

10.11 midi_refpix

The purpose of this technical template is to evaluate the reference positions of the VLTI beams on the MIDI detector for the fine positioning of the target. It also facilitates monitoring and evaluating the stability of the beams inside the MIDI DEWAR.

The reference pixels of MIDI are the two pixels of the detector onto which the centroids of the target images must fall in order to ensure a proper beam overlap. They are measured using the center of a pinhole, in the 3 set-ups OPEN, HIGH_SENS and SCI_PHOT, for telescopes beams A and B feeding MIDI, for the interferometric channels 1 and 2 of MIDI, and for the photometric channels PA and PB of MIDI (in SCI_PHOT only).

10.11.1 Input files

• **REFPIX** *required* set of 5 raw data.

An example of the images is shown in Fig. 10.11.1.

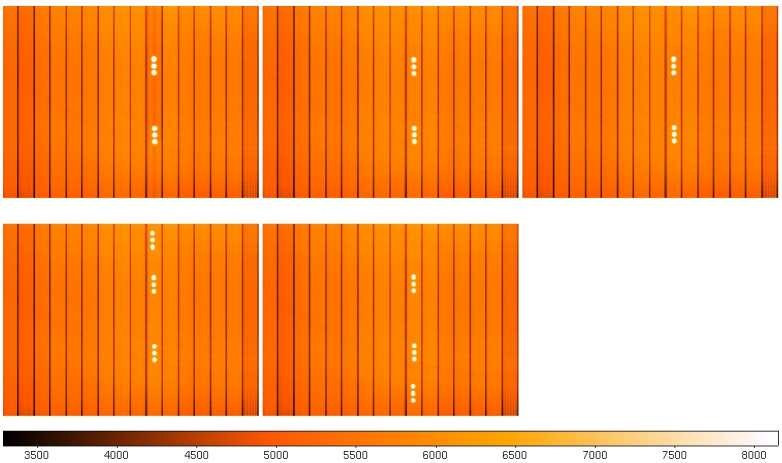

Figure 10.11.1: *Averaged images of the detector region DATA1 for the midi_refpix recipe*

10.11.2 Output files

• **REDUCED_REFPIX** The header of this fits file contains different information on the reference pixels (see QC parameters).

10.11.3 Quality control parameters

A list of the calculated Quality Control parameters, together with as a short description can be found in Appendix C.9.

11 Algorithms

The data reduction procedures applied by the pipeline recipes currently in use (see Section 10) are described here in some detail.

11.1 midi_profile

This recipe derives the 2D profile of the spectral dispersed signal in the SCI_PHOT and HIGH_SENS mode by using a set of AOPEN and BOPEN photometry files (PHOTOM_SP_CALIB/PHOTOM_SP_SCIENCE or PHOTOM_HS_CALIB/PHOTOM_HS_SCIENCE)

The sky subtracted data are co-added and then a two pass threshold is applied in order to get a clean profile of the signal:
In a first pass the standard deviation of the image is derived and all pixels exceeding the latter are marked as pixels belonging to the source signal and not to the background.
In a second pass the standard deviation is recalculated excluding these source-pixels. All pixels of the image not exceeding a certain flux limit (<threshold> times standard-deviation) are set to zero. Finally, the integral of the remaining source profile is normalized to unity perpendicular to the dispersion direction.

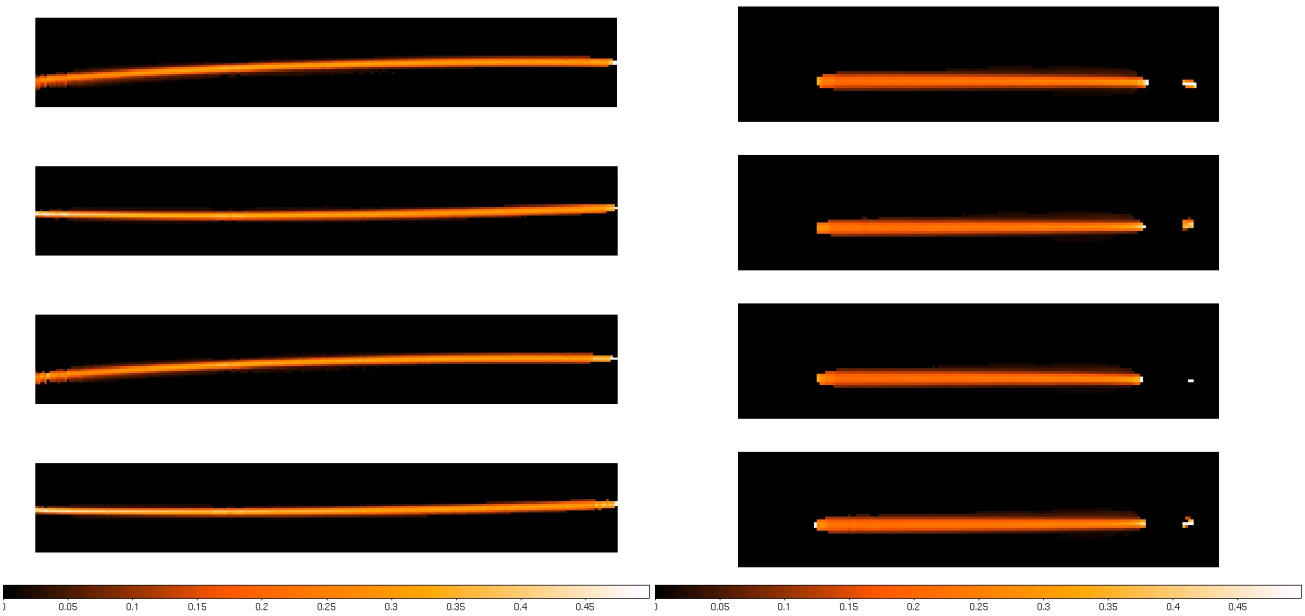

Figure 11.1.1: *Diagnostic files for HIGH SENS GRISM (left panel) and PRISM (right panel) exposures. The normalized profile of the spectrum is clearly visible in the middle of the different detector regions. The black regions have zero values and are used to suppress the background during the signal extraction procedure in various MIDI recipes. From top to bottom: AOPEN DATA1, AOPEN DATA2, BOPEN DATA1, BOPEN DATA2.*

Please note, that the <threshold> variable can be set as a recipe parameter (in sigma units). Moreover, if a signal is present in the AOPEN and BOPEN files, the profile is derived from the sum of the two signals. This increases the SNR of the resulting profile. The diagnostic plots, on the other hand, are saved before the summation has

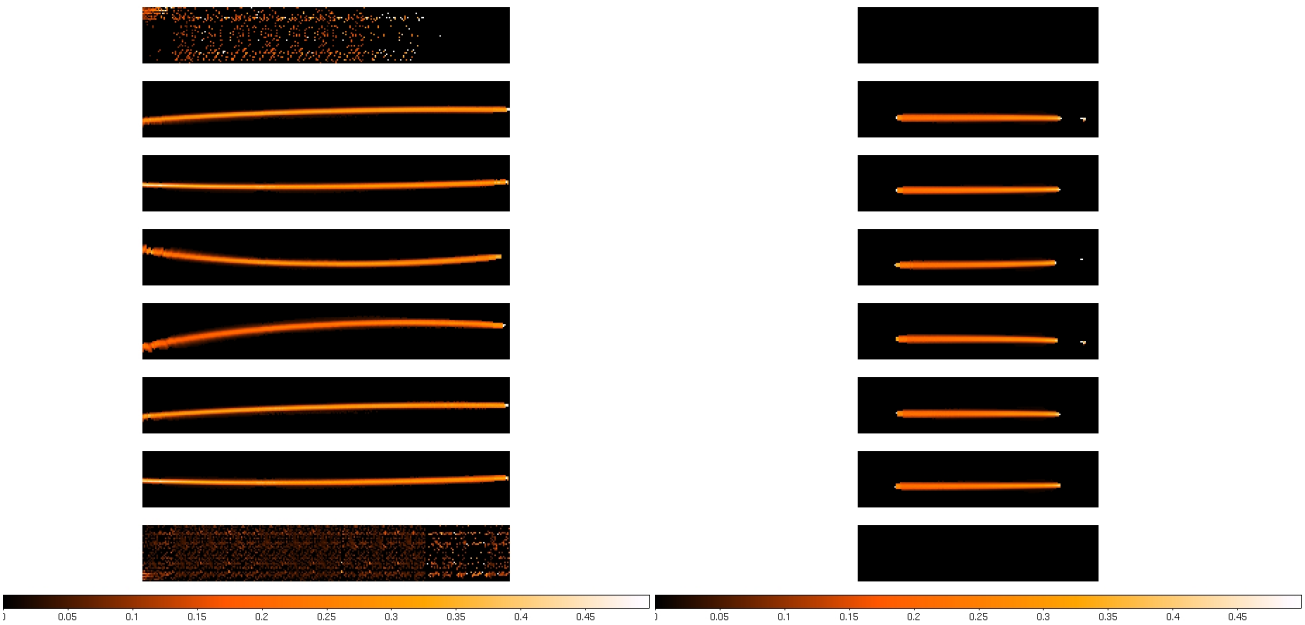

Figure 11.1.2: *Diagnostic files for SCI PHOT GRISM (left panel) and PRISM (right panel) exposures. The normalized profile of the spectrum is clearly visible in the middle of the different detector regions. The black regions have zero values and are used to suppress the background during the signal extraction procedure in various MIDI recipes. From top to bottom: AOPEN DATA1, AOPEN DATA2, AOPEN DATA3, AOPEN DATA4, BOPEN DATA1, BOPEN DATA2, BOPEN DATA3, BOPEN DATA4*

been done. This is more convenient for checking the single contributions.

Typical results of the midi_profile recipe (some diagnostic plots) are shown in Fig. 11.1.1 for the HS case as well as in Fig. 11.1.2 for the SP case. As can be seen from the Fig. 11.1.2 there is no signal in AOPEN DATA1 and BOPEN DATA2. Therefore, this data are not used to derive the main product of the recipe (KAPPA_SP_MASK_(PRISM|GRISM) or KAPPA_HS_MASK_(PRISM|GRISM)).

In order to optimize the SNR for the resulting profile, one should combine as many photometry files as possible. On the other hand, the position of the profile on the CCD could vary with time. Therefore it is usually best to derive the profile OB by OB and after-wards check if the data can be combined. This can be done by blinking the diagnostic files or by comparing the QC parameters, as the QC parameters give the y-position of the profile in different channel intervals (using the already y-normalized profile for the calculation).

An profile analysis of about 150 HIGH_SENS and 70 SCI_PHOT OBs (spread in time over more then one year) with this recipe and by the SExtractor program lead to the following preliminary conclusions:

- The scatter (rms) in the *relative* y-position (y_AOPEN−y_BOPEN) of the profile is around 0.3 pixels in most of the cases. The analysis has been done individually for UTs, ATs, GRISM, PRISM, SCI_PHOT, and HIGH_SENS OBs.
- The scatter (rms) in the y-position of the profile is around 0.3 to 0.5 pixel, but there are also some outliers. The analysis has been done individually for UTs, ATs, GRISM, PRISM, SCI_PHOT, and HIGH_SENS OBs (for DATA1 to DATA4, if present).

- Combining UT and AT data to get a high SNR profile seem to be possible. The visibility results (checked on many calibrator visibilities) are very similar (beside small border effects) if 1) only AT observations are used to derive the profile, 2) only UT observations are used to derive the profile, 3) AT and UT observations are combined to derive the profile.

11.2 midi_kappamatrix

In the SCI_PHOT mode, the photometric and the interferometric channels are observed simultaneously. The beam splitter sends one part of the light to the photometric channels and the other part to the Interferometric channels. The kappa matrix (or transfer ratio) defines the relationship between the intensity in the interferometric channels and in the photometric channels, i.e. it depends on the light splitting ratio.

The midi_kappamatrix recipe derives the kappa matrix in the SCI_PHOT mode following the definition in [10]. It uses a set of AOPEN and BOPEN photometry files as well as the spatial profile of the spectrum (derived by the midi_profile recipe). The spectrum is extracted using the Optimal Extraction algorithm [12] adapted to NIR data, i.e. background dominated images. The errors of the kappa matrix are not calculated but set a priori to 5 per cent.

The SCI_PHOT fits-files contain the photometric and interferometric images in the IMAGING_DATA extension. They are stored in the table columns DATA1, DATA2, DATA3, and DATA4. For convenience, the following nomenclature will be used:

$$\begin{array}{lcl} \mathrm{DATA1} & \rightarrow & \mathrm{P1} \\ \mathrm{DATA2} & \rightarrow & \mathrm{I1} \\ \mathrm{DATA3} & \rightarrow & \mathrm{I2} \\ \mathrm{DATA4} & \rightarrow & \mathrm{P2} \end{array} \tag{1}$$

with P1 (photometry 1), P2 (photometry 2), I1 (interferometri 1), and I2 (interferometri 2).

We define the kappa matrix κ as follows:

$$\begin{pmatrix} \mathrm{I1} \\ \mathrm{I2} \end{pmatrix} = \begin{pmatrix} \kappa_{11} & \kappa_{12} \\ \kappa_{21} & \kappa_{22} \end{pmatrix} \begin{pmatrix} \mathrm{P1} \\ \mathrm{P2} \end{pmatrix} \tag{2}$$

leading to the equationary system:
I1=κ_{11} P1 + κ_{12} P2
I2=κ_{21} P1 + κ_{22} P2
where for simplicity reference to wavelength index is dropped

If the kappa matrix is known, the photometric flux in I1 and I2 can be derived from P1 and P2 in an interferometric observation, and normalized visibilities can be calculated.

Please note, that the matrix coefficients are a function of wavelength but for simplicity the reference to the wavelength index is dropped.

In order to derive the kappa matrix a set (typically 3000 to 5000) of AOPEN (telescope/beam A is open) and BOPEN (telescope/beam B is open) frames are taken. As can bee seen from Fig.11.1.2, AOPEN files have no flux in P1, but the flux is splitted between I1, I2, and P2. On the other side, BOPEN files have no flux in P2,

but the flux is splitted between P1, I1, and I2. Therefore the coefficient of the kappa matrix can be derived as follows:

$$\begin{aligned} \kappa_{11} &= \frac{\mathrm{I1}}{\mathrm{P1}} \quad \text{from BOPEN files} \\ \kappa_{12} &= \frac{\mathrm{I1}}{\mathrm{P2}} \quad \text{from AOPEN files} \\ \kappa_{21} &= \frac{\mathrm{I2}}{\mathrm{P1}} \quad \text{from BOPEN files} \\ \kappa_{22} &= \frac{\mathrm{I2}}{\mathrm{P2}} \quad \text{from AOPEN files} \end{aligned} \tag{3}$$

The following steps are executed by the recipe to extract the kappa matrix:

1. Reading chopped files and subtracting the sky. Each individual sky and target patch is averaged and then the the sky patches are subtracted from the target patches (one after another).
2. Extracting the signal using Horne 1986 [12] and assuming that the frames are sky-dominated, i.e. $V(x) \rightarrow V_0 = const.$ in [12]. The used source profile was previously derived by the midi_profile recipe.
3. Deriving the kappa matrix as described in eqn. (3).

An example of the kappa matrix is shown in Fig. 11.2.1 for GRISM and in Fig. 11.2.2 for PRISM observations.

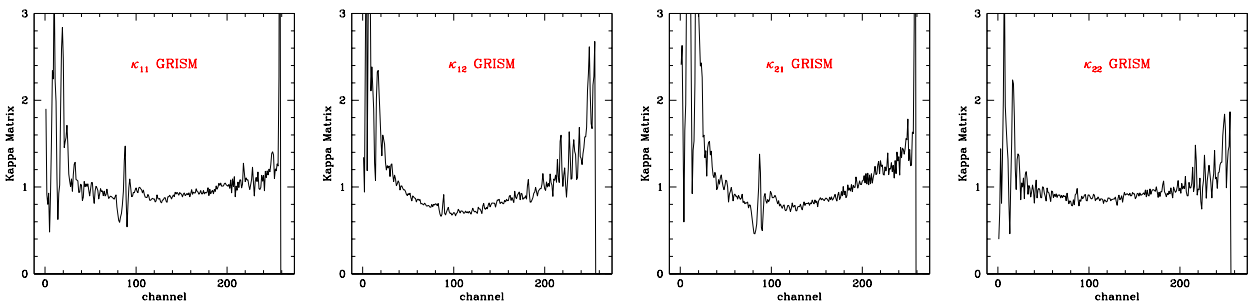

Figure 11.2.1: *Kappa matrix for GRISM observations*

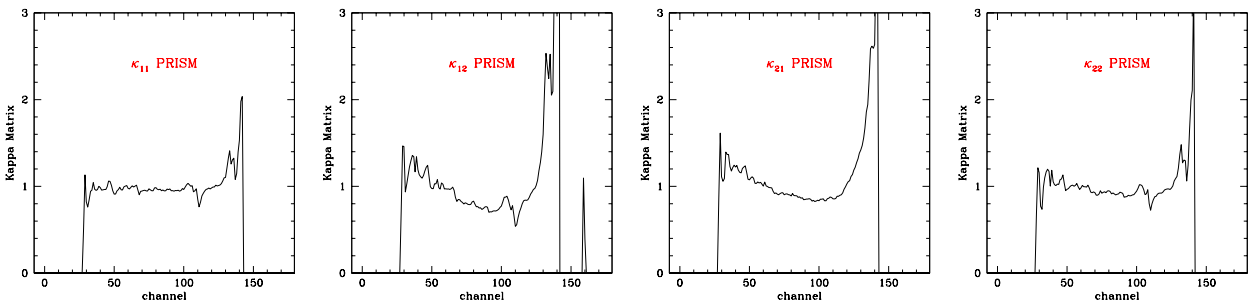

Figure 11.2.2: *Kappa matrix for PRISM observations*

11.3 midi_intopd

The midi_intopd recipe monitors the stability of the internal OPD (path difference) of MIDI using the group delay as a tracer. As described in e.g. Lawson 1995 [13], for Edser-Butler fringes (fringes of equal chromatic order) the path difference can be monitored by the observation of the number of fringes in a given wavelength/frequency interval. If p fringes are counted between wavelength λ_{min} and λ_{max}, the path difference x is given by

$$x = \frac{p}{\Delta\kappa} \tag{4}$$

where

$$\Delta\kappa = \frac{1}{\lambda_{min}} - \frac{1}{\lambda_{max}} \tag{5}$$

As the fringe signal in the data is very strong (Lamp), the signal can be extracted by a simple zero-unity mask and no Optimal-extraction algorithm has to be applied (the position of the rectangular mask can be controlled by the recipe parameters). After multiplying the data with this mask thus eliminating most of the thermal background, the data is collapsed perpendicular to the dispersion direction. As the two interferometric windows I1 and I2 are shifted by $\lambda/2$, subtracting them from each other results in a clean fringe signal (background is not interfering and therefore automatically removed).

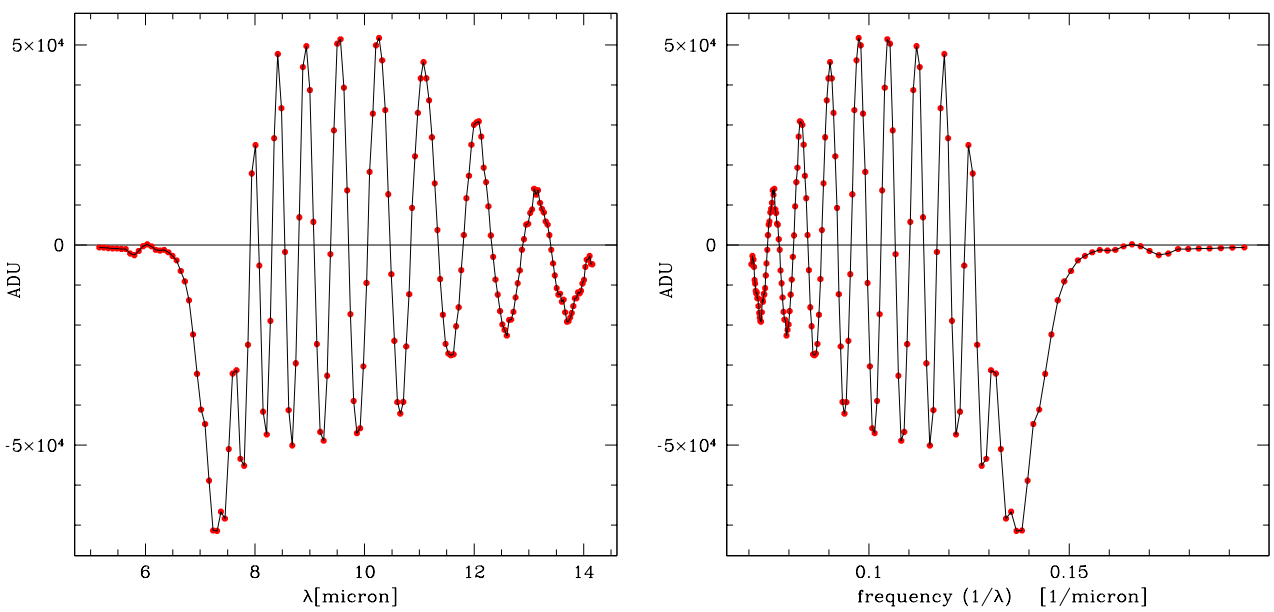

Figure 11.3.1: *Fringe signal as a function of wavelength (left panel) and of frequency (right panel) of a single observation. The measurements in the various channels on the detector (red dots) are connected by black lines to guide the eye*

Fig. 11.3.1 shows a typical extracted fringe signal of a single exposure as a function of wavelength and frequency. Typically 1000 successive single exposures are taken in one OB and analyzed by the recipe in order to monitor the time dependency of the internal path difference. Fig. 11.3.1 clearly shows two important points:

1. The spacing of the fringes is regular in the frequency domain but not in the wavelength domain.

2. The fringe is not equally sampled in wavelength and/or frequency space as there is no linear dependency between channel (pixel) and λ.

As a FFTW analysis implies equally sampled data, the signal would have to be resampled onto a new grid. On the other hand resampling a fast oscillating signal could introduce reduction artifacts. In order to avoid resampling and flux reshuffling the flux of the signal we decided to use the algorithm described in Scargle 1982 [11]. The paper studies the reliability and efficiency of detection with the most commonly used technique, the periodogram, in the case where the observation times are unevenly spaced. Scargle modified the classical definition of the periodogram analysis in order to retain the simple statistical behavior of the evenly spaced case. With this modification, periodogram analysis and least-squares fitting of sine waves to the data are exactly equivalent.

Although the paper deals with (periodic) signals in the time domain, it can easily be adapted to analyze a fringe signal in the frequency domain and thus to determine the path difference. The path difference stored in the recipe product is derived from the peak of the periodogram without any further interpolation (e.g. by a spline). Moreover, as the absolute value of the power of the periodogram is not of much interest for the moment, the latter is not yet validated.

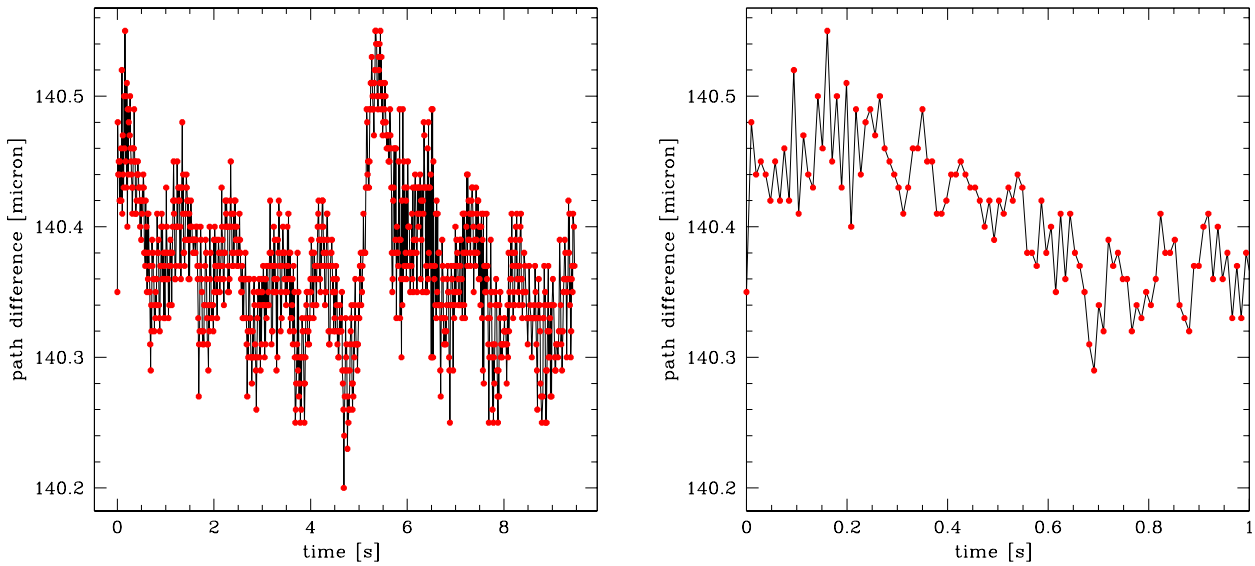

Figure 11.3.2: *Path difference variation derived by the midi_intopd recipe within 10 seconds (left panel) and 1 second (right panel)*

The typical result of the midi_intopd recipe can be found in Fig. 11.3.2. The mean path difference for this OB is $x = 140.37 \pm 0.05$ micron over about 10 sec. Please note, that the observation has been executed with the piezo not scanning, i.e. the OPD should not move during the observation. This implies, that the internal stability of the MIDI OPD is in the order of 0.05 micron.

11.4 midi_fringe_all: HIGHSENS observations

This section outlines the algorithm for the midi_frine_all recipe if the observations are taken in the HIGHSENS mode. The algorithm consists of the following major components:

- Data Compression
- Photometry Computation
- Fringe Estimation
- Visibility Computation
- Transfer Function and Calibrated Visibility Computation (optional)

The above algorithm components that are incorporated in the MIDI pipeline are described in the following sections.

11.4.1 Data Compression

Each input detector data consists of n scans. Each scan consists of f frames and each frame is essentially an image of XY dimension. The process of data compression in DISPERSED processing mode entails collapsing the data in spatial Y direction for each frame. The collapsed data that provides the averaged flux for each wavelength λ, and for each frame is given by:

$$X_k^j(\lambda) = \sum_i w(i)x(i,j,k,\lambda) \tag{6}$$

where $x(i,j,k,\lambda)$ is the input detector data for frame number k in detector region j and wavelength λ. i refers to each pixel within that region. For region j there is a weighting mask which assigns a weight $w(i)$ to each pixel.

The combined raw data is obtained from the two interference channels in use. That is $j = I_1$ and $j = I_2$ using a weighted subtraction:

$$X_k^c(\lambda) = K_1 X_k^{I1}(\lambda) - K_2 X_k^{I2}(\lambda) \tag{7}$$

In this version we simply use $K_1 = K_2 = 1$.

11.4.2 Photometry computation

It is essential for an optimal background subtraction, that the sky file is processed in the same way as the interferometric file. We denote the collapsed background thereby determined for region j as $B(j)$, where for simplicity reference to wavelength index is dropped. Thus:

$$B_k^j(i) = \sum_i w(i)x(i,j,k)^{(sky)} \tag{8}$$

For photometric purposes, we shall consider that there is no temporal information within the sequence of frames in a sky file, so we shall average the results from each of k frames to obtain an estimate of the collapsed background level for each region:

$$\bar{B}^j = \frac{1}{k}\sum_{k=0}^{k-1} B_k^j \tag{9}$$

The appropriate photometric level $P(j)$ for each detector region j is determined from the average DC level of the collapsed signal, minus the average level of similarly determined sky file.

$$\bar{P}^j = \frac{1}{k}\sum_{k=0}^{k-1} X_k^j - \bar{B}^j \tag{10}$$

Division of the average correlated flux by this photometric level is required in order to obtain a visibility determination which is independent of the source flux. In interferometric measurements using a clear aperture in the image plane (rather than a pinhole) there is no apparent reason for a true drift in the photometric level during a short observation. Thus we shall employ, in this case, the $P(j)$ determined from the entire interferometric observation, even when the interferometry is limited to a subset of that observation. When pinholes are to be employed, it will become necessary to tailor the photometric averaging window to match that over which correlated flux was measured.

It is reasonable to normalize the results of correlated flux in interference channels 1 and 2 using the photometry determined from that channel alone, i.e. P^{I1} and P^{I2} respectively. This will reduce any effect due to mismatches between the weighting masks in the two regions (which is inevitable since there will generally not be a one to one match between pixels) interacting with pointing errors.

The appropriate photometric level to normalize the visibility obtained from the "combined" interferometric data sequence X_k^c, should best be a mixture of the photometry determined in each interferometric region separately, perhaps using the same weighting coefficients K_1 and K_2 although this choice is not critical. Thus we choose:

$$P^c = K_1 P^{I1} + K_2 P^{I2} \tag{11}$$

Photometric normalization of correlated flux obtained in pinhole modes will face substantially different considerations, whether using photometry derived from the interferometric channels or from the photometric pickoff channels. This is because the photometry will now be sensitive to atmospheric wave-front errors. Photometry in the case of observation of fields which are not totally unresolved by the UT's will face further complications. These matters are not addressed in the current version of the software. Here we only compute visibilities from the I_1, I_2, and combined interference channels using the respective photometric determinations obtained employing identical weightings.

11.4.3 Fringe Estimation

The collapsed data for each frame in the two interferometric channels (and the combined interferometric data), X^{I1}, X^{I2} and X^c, are organized into scans, according to the applied OPD schedule. Let us denote the data organized into scans as $y_i(t)$ where we shall no longer identify the source as being from one or another interference channel; the same processing will apply to all. Here i refers to the scan number and t is an integer referring to the frame within that scan. Thus, if there are M frames per scan, then frame number k will get mapped to $y_i(t)$ with $i = (int)k/M$ and $t = mod_m(k)$.

The DC level of the raw data from scan i is found by:

$$\bar{y}_i = \frac{1}{M} \sum_{t=0}^{M-1} y_i(t) \tag{12}$$

This DC level is subtracted from each element of y, and placed into a larger array y' of length N, by padding the data with zeros:

$$y'_i(t) = y_i(t) - \bar{y}_i \qquad (0 \leq t < M) \tag{13}$$

$$y'_i(t) = 0 \qquad (M \leq t < N) \tag{14}$$

The data from each scan is then Fast Fourier Transformed into frequency space, with the frequency index being denoted by f

$$Y_i(f) = FT(y'_i) \tag{15}$$

The Fourier transforms from each scan, Y_i, are combined incoherently by averaging the squared magnitudes of the individual scans. Only positive frequencies $(0 < f < N/2)$ need be considered. The scan range i_1 to i_2 can include the full range of scans, or a subset thereof.

$$S(f) = \frac{1}{i_2 - i_1 + 1} \sum_{i=i_1}^{i_2} \mid Y_i(f) \mid^2 \tag{16}$$

An estimate of the noise level at each frequency, $S(f)^{(noise)}$ is subtracted from each spectral channel to obtain an estimate of the power in that channel due to the signal alone. Note that this result is allowed to be negative.

$$S'(f) = S(f) - S(f)^{(noise)} \tag{17}$$

11.4.4 Visibility Computation

Using the interference signal power spectral density thus obtained, an estimate of the total correlated power (that is signal power, not optical power) is found by summing S' over a frequency range which corresponds to the wavelength range of the expected optical signal, broadened due to atmospheric OPD noise and truncation of

the raw scan data in delay space. Consider that we have set the desired frequency range to f_1 to f_2. Then the *unnormalized* visibility estimate is given by:

$$v^2 = A \sum_{f=f_1}^{f_2} S'(f) \tag{18}$$

In order to obtain a *normalized* visibility estimate, we must divide the above by the square of the photometry for the interference signal (detector region) in question, obtained as described above.

$$V^2 = \frac{v^2}{P^2} \tag{19}$$

The constant A can, in principle, be adjusted to obtain a calibrated visibility determination. In theory, if there were no other sources of visibility loss, and in the presence of a perfect balance between the photometric contribution from each telescope, then A could be found from the optical spectrum of the received flux. In the case of a uniform spectrum with a boxcar optical filter whose frequency range (as translated to the indices of the spectra determined using the above procedures), ranges from f_{lo} to f_{hi}, the exact choice for A would be $1/(f_{hi} - f_{lo})$. However in practice, A can never be accurately determined a priori. Moreover, the visibilities output from any algorithm will ultimately need to be calibrated using stellar sources, in order that all sources of visibility loss are accounted for. Thus any possible choice of A will cancel as soon as actual calibrated visibilities on the sky are obtained. However results will be affected between observations in which the optical frequency calibration, that is the actual optical frequency (wavelength) referred to by the frequency index f, is changed, either due to changes in the scanning OPD step, or due to changing the array size N in which the FFT is performed. This could be taken into account, for instance, by setting $A = 1/f_{THz}A$ where f_{THz} is the (non integer) representation in f units, of an optical frequency of 1 THz. Such a scheme would enable comparison of V^2 determinations made under different optical frequency calibrations. However cross-calibration of observations performed using different optical spectral filters, will not be easily possible.

It is also possible to make a direct determination of the actual visibility contrast (including all the pre-detection sources of visibility loss) by employing the approximation that the expected value of the square root of x is equal to the square root of the expected value of x. Spectral channels for which this approximation breaks down are handled separately by first summing their power and then treating them as a single spectral channel, with a statistical correction factor. Channels are considered "hot" if they have a signal plus noise power level which exceeds the expected noise power in that channel by a factor of T^2 where we currently employ a threshold value $T = 2$. In a strong observation there will be very little contribution from the "cold" channels, which do not achieve this threshold.

Again we limit the effect of out of band noise by limiting the frequency range considered to f_1 to f_2. Then the *unnormalized* correlated flux in detector units is found by integrating the amplitude obtained from each spectral channel by taking the square root of the estimated signal power $S'(f)$, which we computed by taking the PSD and subtracting the expected noise contribution.

$$v = \sum_{f=f_1}^{f_2} \sqrt{S'(f)} \tag{20}$$

The subtle correction regarding the contribution of the "cold" channels, is not currently described.

Again, we can obtain a *normalized* visibility by dividing the correlated flux determination by the photometric determination P described above.

$$V = \frac{v}{P} \tag{21}$$

The result, of course, is still subject to calibration on the sky in order to account for the visibility losses which are not otherwise observable. However it is a true measure of the optical fringe contrast observed at the detector.

11.4.5 Transfer Function and Calibrated Visibility Computation (optional)

In order to compute the transfer function (TF) the parameters of the observed calibrator target are compared with those given in the calibrator database. If a close match is found its diameter together with the baseline information are used to compute the theoretical (expected) visibility. The transfer function is then the ratio of the un-calibrated visibility and the expected visibility.

If a transfer function can be found in a SOF, the recipe midi_fringe_all automatically computes *Calibrated Visibilities* as follows: $V_{\text{calibrated}} = V_{\text{measured}} \times \text{TF}$.

11.5 midi_fringe_all: SCIPHOT observations

This section outlines the algorithm for the midi_fringe_all recipe if the observations are taken in the SCIPHOT mode.

The algorithm for the SCIPHOT mode closely follows the one from the HIGHSENS. The main difference arises from the way the raw data are structured and taken. In HIGHSENS, the photometry files are taken closely after the interferometry file, whereas in SCIPHOT mode the interferometric and the photometric flux is taken simultaneously. Moreover, the interferometric and photometric data are chopped (in HIGHSENS only the photometric data is chopped). This allows a better sky subtraction for both signals.

As described in the HIGHSENS mode, in order to derived normalized uncalibrated visibilities the photometry of the object has to be known (see Sec. 11.4.2 and Sec. 11.4.3). For HIGHSENS observations dedicated photometric observations are carried out after the interferometric observation, but for SCIPHOT data interferometric and photometric information are collected simultaneously. Therefore, one has to extrapolate the photometry in the two interferometric channels I1 and I2. This is done by the usage of the (measured) photometric data in P1 and P2 together with the kappa matrix (see Sec. 11.2) as follows:

I1=κ_{11} P1 + κ_{12} P2
I2=κ_{21} P1 + κ_{22} P2

The recipe calculating the kappa matrix is described in section 10.2 whereas the algorithm can be found in section 11.2.

11.6 midi_acq

This recipe derives the position, the size as well as the flux of the observed source. The algorithm consists of the following components:

- Data compression
- Sky (background) removal
- Computation of the relevant parameters

For each detector region all image-frames are averaged to one single image. This is done for both $Target$ as well as Sky frames excluding the $Undefined$ frames. The Sky is then subtracted from the $Target$ to provide the true $Signal$ image. The position and size of the target is calculated by fitting a a 2D Gaussian profile to the source and the the flux is derived by integrating the source pixels.

11.7 midi_wavecal

The purpose of this recipe is the calibration of the detector channels, i.e. to assign to each detector pixel the corresponding wavelength. In the current version of the software, wavelength calibration is carried out using the available data files for the following narrow band filters:

- ArIII
- NeII
- SIV

As depicted in section 10.7, each data file is averaged with background removed. The resulting images emerge with three pinholes. The position of the central pinholes are accurately computed to provide three spatial coordinates. These coordinates are then related to the following manufacturers parameters:

Filter	Peak Frequency (Hz)	Peak Wavelength (micron)
NeII	2.34297e+13	12.7955
SIV	2.85818e+13	10.489
ArIII	3.33005e+13	9.00272

The three pairs of parameters listed above provide all information for a linear fit, from which the detector channels can be calibrated.

11.8 midi_detlin

This recipe evaluates the relationship between the detector counts and the number of photons

It uses a rectangular area $(x1, y1, dx1, dy1)$ in the center of the image for all files. Each pixel within this window is then characterized over the entire integration time given by the number of exposures. This involves computing the coefficients of a 3rd order polynomial which then describes the flux versus integration time for all the pixels in the sub-window. The input data for each pixel is given by:

$$D = Array[t_e(e), \hat{p}(e)] \tag{22}$$

where
e = 1, 2, ... n
n = number of exposures
t_e = exposure time for each file

$\hat{p}$ beeing the mean pixel flux in each exposure e given by:

$$\hat{p}(e) = \sum_{f=1}^{F} p(e) \tag{23}$$

where
F = the number of frames in each exposure
$p(e)$ = flux for each pixel in the rectangular area $(x1, y1, dx1, dy1)$

The coefficients of the polynomial functions that describe the above relationships for each pixel is given by:

$$y = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{24}$$

Additionally a regression test is also carried out for each pixel array of eq. (22). From this the following products are created:
- A map of the mean standard deviations describing the deviation of each pixel from linearity.
- An overall mean standard deviation for the window.

11.9 midi_detron

The purpose of this recipe is to evaluate the read-out noise of the detector by analyzing a large amount of BIAS frames (usually more then 1000) taken by the observation template.
The recipe computes the standard deviation for each pixel of the MIDI detector, thus creating a product image given by the following n x m matrix:

$$IMAGE_{detRon} = |\sigma(i,j)| \tag{25}$$

where
i = 1, 2, ... n with n beeing the x-dimension of the image ,
j = 1, 2, ... m with m beeing the y-dimension of the image
$\sigma(i,j)$: standard deviation of pixel (i,j)

11.10 midi_dsptrn

This recipe measures the transmission characteristics of the dispersive elements (Prism and Grism) in order to monitor the evolution of their coatings with time.

The algorithm includes the following stages:

- For each file determine the coordinates of the target $(x1, y1, dx1, dy1)$ using a 2D Gaussian fit
- Compute mean target flux with background removed
- Get the corresponding integration time
- For each file compute the integral count $P_s(f)$ and $P_p(f)$, f = 1, ... 6
- Dispersive Transmissions are then obtained from the ratio of the corresponding integral count as shown below

For each pair of files the Dispersive Transmission is given by:

$$\alpha(i) = P_s(j)/P_p(k) \tag{26}$$

where
n = the number of data files
i = 1, 2, ... n/2
j = (n - n/2), (n - n/2) + 1, ... n
k = 1, 2, ... n/2
P_s is the integral count for the spectroscopy data file
P_p is the integral count for the photometry data file.

The above formulation applies to both Prism as well as Grism. The integral count in all cases is given by:

$$P_s(j) = \frac{\phi_s}{T_s} \tag{27}$$

$$P_p(k) = \frac{\phi_p}{T_p} \tag{28}$$

where
ϕ_s = mean flux per pixel for the spectroscopy data
ϕ_p = mean flux per pixel for the photometry data
T_s = total exposure time for the spectroscopy data
T_p = total exposure time for the photometry data.

The mean flux per pixel in both cases are given by

$$\phi_s = \frac{1}{FC_s} \mid \sum_{f=1}^{F} \Phi_s(f) \mid \tag{29}$$

$$\phi_p = \frac{1}{FC_p} \mid \sum_{f=1}^{F} \Phi_p(f) \mid \tag{30}$$

where

F = number of frames

Φ_s = spectroscopy's total flux in the image sub-window (x1, y1, dx1, dy1)

Φ_p = photometry's total flux in the image sub-window (x1, y1, dx1, dy1)

C_s = spectroscopy's count of the sub-window (x1, y1, dx1, dy1) containing the image

C_p = photometry's count of the sub-window (x1, y1, dx1, dy1) containing the image

11.11 midi_refpix

This recipe evaluates the reference positions of the VLTI beams on the MIDI detector for the fine positioning of the target. It also facilitates monitoring and evaluating the stability of the beams inside the MIDI DEWAR.

Following the initial file classification and format analysis the data from each FITS file is compressed along time. The result is five image files. Each image file represents the averaged image of the detector region. Three types of images corresponding to three types of beam combiner are analyzed. These beam combiners are:

- File 1: OPEN
- File 2: HIGHSENS
- File 3: HIGHSENS
- File 4: SCHI PHOT
- File 5: SCHI PHOT

In order to locate the target coordinates a 2D Gaussian fit algorithm is used with a given search size and expected pinhole position. For beam combiner OPEN and HIGHSENS two beams are considered and for the SCIPHOT beam combiner the parameters for three beams are derived.

A Installation

See: Installation instructions

B Abbreviations And Acronyms

B.1 Abbreviations

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
AO	Adaptive Optics
ASTO	Archive Storage System
AT	Auxiliary Telescope (1.8m)
CalibDB	Calibration Database
CPL	Common Pipeline Library
CS	Constraint Set
DFS	Data Flow System
DHS	Data Handling System
DICB	Data Interface Control Board
DID	Data Interface Dictionary
DIT	Detector Integration Time
DAS	Data Analysis Software
DFS	Data Flow System
DMD	Data Management and Operations Division
DO	Data Organizer
DPD	Data Products Department
DRS	Data Reduction Software
ESO	European Southern Observatory
ESO/MIDAS	ESO's Munich Image Data Analysis System
ETC	Exposure Time Calculator
FITS	Flexible Image Transport System
FOV	Field Of View
FSU	Fringe Sensor Unit
GUI	Graphical User Interface
HDU	Header Data Unit of a FITS file
ICD	Interface Control Document
ISS	VLTI Supervisor Software
MIDI	Mid-Infrared interferometric instrument
MIR	Mid Infra Red
OB	Observation Block
OD	Observation Description
OPC	Observing Program Committee
OPD	Optical Path Difference
OPL	Optical Path Length
OS	Observation Software
OT	Observation Toolkit
PAF	VLT-Parameter File
P2PP	Phase 2 Proposal Preparation

PRIMA	Phase-Referenced Imaging and Microarcsecond Astrometry
QC	Quality Control
QC1	Quality Control Level 1
RB	Reduction Block
RBS	Reduction Block Scheduler
RTD	Real Time Displayer
SM	Service Mode
SOF	Set Of Frame
STRAP	System for Tip-Tilt Removal with Avalanche Photodiodes
TSF	Template Signature File
USG	User Support Group
UT	Unit Telescope of VLTI
UTC	Coordinated Universal Time
VINCI	VLT Interferometer Commissioning Instrument
VLT	Very Large Telescope
VLTI	Very Large Telescope Interferometer
VM	Visitor Mode

B.2 Glossary

Calibration Database: Database containing master calibration data.

Data Organizer: A DFS component which classifies and analyses the content of any incoming raw frame and creates the corresponding Reduction Block. Assembles calibration frames and raw data to be processed following data reduction recipes (data reduction procedures) specified in a RB.

Exposure: A synonym for the acquisition of a single data frame, typically resulting in a single FITS file.

Observation: A coordinate sequence of telescope, instrument, and detector actions that results in a scientific or technical dataset.

Observation Block: Smallest observational unit within the Data Flow System. It contains a sequence of high level operations, called *template*, that need to be performed sequentially and without interruption in order to ensure the scientific usefulness of an observation. Observation Blocks may contain scheduling requirements. They are used both in Visitor and Service Mode to acquire data.

Pipeline product: Result of the execution of a Reduction Block

Quality Control level 0: On-Line tool that checks whether Service Mode OBs have been executed under the conditions specified by the astronomer. QC0 is executed on raw data.

Quality Control level 1: QC1 consists of quality checks on pipeline processed data. The QC1 parameters are used to assess the quality of calibration products and the performance of the instrument.

Reduction Block: A Reduction Block is an ASCII file containing all the relevant information to do a pipeline data reduction. It indicates the observing instrument, the data reduction recipe to be executed, the pipeline product file prefix, its full path, input raw files and their identification assigned by the DO, input reference and calibration files with their classification assigned by the DO. Each file name must appear with its complete path

Reduction Block Scheduler: A tool which schedules and executes RBs created and sent by the DO. RBS sends the RB to the DRS which will actually perform the data reduction

Reduction pipeline: Subsystem of the DFS in charge of pipeline processing. Applies reduction recipes and its parameters (calibration frames) on raw frames to generate pipeline products.

Reduction recipe: Standard procedure for reducing observational data in a standard way. Recipes are implemented for each of the instrument templates. Those scripts take as input raw frames and execute them in a particular Data Reduction System (DRS).

Service Mode: Observing operations mode where the astronomer submits a detailed description of their observing program to ESO for later possible execution. Service Mode programs are executed primarily in order of their OPC assigned priority but only when the astronomer specified observing conditions are achieved on site.

Template: A high-level data acquisition operation. Templates group commonly used procedures into well defined and standardised units. They can be used to specify a combination of detector, instrument, and telescope configurations and actions. Templates have input parameters described by a template signature, and produce results that can serve as input to other templates.

Visitor Mode: Observing operations mode where the astronomer is present at the telescope when the observing program is being executed.

C QC1 Dictionary

The following tables give a list of the Quality Control parameters of the various recipes.

C.1 QC parameter of the midi_profile recipe

QC Parameter	Short Description
QC XXXX CENT BINNED1	y position of the profile centroid in bin1
QC XXXX CENT BINNED2	y position of the profile centroid in bin2
QC XXXX CENT BINNED3	y position of the profile centroid in bin3
QC XXXX CENT BINNED4	y position of the profile centroid in bin4
QC XXXX CENT BINNED5	y position of the profile centroid in bin5
QC XXXX MAX BIN1	maximum of the profile in bin1
QC XXXX MAX BIN2	maximum of the profile in bin2
QC XXXX MAX BIN3	maximum of the profile in bin3
QC XXXX MAX BIN4	maximum of the profile in bin4
QC XXXX MAX BIN5	maximum of the profile in bin5
QC XXXX MAXSUM BIN1	maximum of the x-collapsed profile in bin1
QC XXXX MAXSUM BIN2	maximum of the x-collapsed profile in bin2
QC XXXX MAXSUM BIN3	maximum of the x-collapsed profile in bin3
QC XXXX MAXSUM BIN4	maximum of the x-collapsed profile in bin4
QC XXXX MAXSUM BIN5	maximum of the x-collapsed profile in bin5

where XXXX is Y1 or Y2 or Y3 or Y4 and determines the corresponding DATA1 to DATA4 measurement. Please note that HS has only Y1 and Y2 whereas SP runs up to Y4 The values are calculated in the same channels as for the fringes (BINNED1 to BINNED5).
The recipe determines the y-position of the centroid in the various bins as well as the maximum (MAX) and the maximum of the x-collapsed bins (MAXSUM) from the already y-normalized profile.
The higher the measured maximum the more concentrated the profile. If the observations e.g. oscillates, the spectral profile gets more flux in the wings, thus decreasing the maximum (as the integral of the single spectral channel is normalized to unity in the y-direction).

C.2 QC parameter of the midi_kappamatrix recipe

‘ QC Parameter	Short Description
QC KAPPA MEDIAN 11	Median value of the kappa matrix 11 in a predefined λ-interval
QC KAPPA STDEV 11	Standard deviation of the kappa matrix 11 in a predefined λ-interval
QC KAPPA MEDIAN 12	Median value of the kappa matrix 12 in a predefined λ-interval
QC KAPPA STDEV 12	Standard deviation of the kappa matrix 12 in a predefined λ-interval
QC KAPPA MEDIAN 21	Median value of the kappa matrix 21 in a predefined λ-interval
QC KAPPA STDEV 21	Standard deviation of the kappa matrix 21 in a predefined λ-interval
QC KAPPA MEDIAN 22	y Median value of the kappa matrix 22 in a predefined λ-interval

QC KAPPA STDEV 22	Standard deviation of the kappa matrix 22 in a predefined λ-interval

The median value as well as the standard deviation of the kappa matrix is derived in the following channel range: x=[140,160] ($\bar{\lambda} \sim 11\ \mu m$) for GRISM and x=[65,85] ($\bar{\lambda} \sim 11.4\ \mu m$) for PRISM. The range has been chosen to be small, as there is usually a gradient in the overall kappa matrix which would systematically increase the standard deviation for a larger intervall.

C.3 QC parameter of the midi_intopd recipe

QC Parameter	Short Description
QC INTOPD MIN	minimum OPD offset
QC INTOPD MAX	maximum OPD offset
QC INTOPD MEAN	mean OPD offset
QC INTOPD MEDIAN	median OPD offset
QC INTOPD STDEV	standard deviation of the OPD offset

All quantities are measured in meters

C.4 QC parameter of the midi_fringe_all recipe

QC Parameter	Short Description
QC TARTYPE INTERF CHANGED	Number of type-modified interferometric frames
QC TARTYPE PHOTOMA CHANGED	Number of type-modified photometric-beam-A frames
QC TARTYPE PHOTOMB CHANGED	Number of type-modified photometric-beam-B frames
QC AT117	Applied Transfer Function
QC AT127	Applied Transfer Function
QC AT232	Applied Transfer Function
QC AT43	Applied Transfer Function
QC AT63	Applied Transfer Function
QC AT91	Applied Transfer Function
QC AV DPHOTA	Error of Average photometry beam A
QC AV DPHOTB	Error of Average photometry beam B
QC AV DPHOTI	Error of Average photometry Interferometric beam
QC AV DPHOTN	Error of Average net photometry
QC AV PHOTA	Average photometry beam A
QC AV PHOTB	Average photometry beam B
QC AV PHOTI	Average photometry beam I
QC AV PHOTN	Average net photometry
QC BL ANGLE	Baseline vector angle
QC BL LENGTH	Baseline vector length
QC BL UVW1	U component of UVW vector
QC BL UVW2	V component of UVW vector

QC BL UVW3	W component of UVW vector
QC CALIB DDIAM	Variance of Calibrator Diameter
QC CALIB DEC	Calibrator Declanation
QC CALIB DIAM	Calibrator Diameter
QC CALIB DIST	Calibrator Radial Distance
QC CALIB FLAG	Calibrator Quality Flag
QC CALIB FLUX	Calibrator N-band flux
QC CALIB NAME	Name of the calibrator entry
QC CALIB POS	Position of the calibrator entry
QC CALIB RA	Calibrator Right Ascension
QC CALV117	Calibrated Visibility
QC CALV127	Calibrated Visibility
QC CALV232	Calibrated Visibility
QC CALV43	Calibrated Visibility
QC CALV63	Calibrated Visibility
QC CALV91	Calibrated Visibility
QC CHPROCESSED	Number of channels processed
QC CHREJECTED	Number of channels rejected
QC COT117	Inverse Transfer Function
QC COT127	Inverse Transfer Function
QC COT232	Inverse Transfer Function
QC COT43	Inverse Transfer Function
QC COT63	Inverse Transfer Function
QC COT91	Inverse Transfer Function
QC DAT117	Variance of Applied Transfer Function
QC DAT127	Variance of Applied Transfer Function
QC DAT232	Variance of Applied Transfer Function
QC DAT43	Variance of Applied Transfer Function
QC DAT63	Variance of Applied Transfer Function
QC DAT91	Variance of Applied Transfer Function
QC DCALV117	Variance of Calibrated Visibility
QC DCALV127	Variance of Calibrated Visibility
QC DCALV232	Variance of Calibrated Visibility
QC DCALV43	Variance of Calibrated Visibility
QC DCALV63	Variance of Calibrated Visibility
QC DCALV91	Variance of Calibrated Visibility
QC DCOT117	Variance of Inverse Transfer Function
QC DCOT127	Variance of Inverse Transfer Function
QC DCOT232	Variance of Inverse Transfer Function
QC DCOT43	Variance of Inverse Transfer Function
QC DCOT63	Variance of Inverse Transfer Function
QC DCOT91	Variance of Inverse Transfer Function
QC DT117	Variance of Measured Transfer Function
QC DT127	Variance of Measured Transfer Function
QC DT232	Variance of Measured Transfer Function

QC DT43	Variance of Measured Transfer Function
QC DT63	Variance of Measured Transfer Function
QC DT91	Variance of Measured Transfer Function
QC DTHEOVIS	Variance of Theoretical Visibility
QC DUNCALV117	Variance of Uncalibrated Normal Visibility
QC DUNCALV127	Variance of Uncalibrated Normal Visibility
QC DUNCALV232	Variance of Uncalibrated Normal Visibility
QC DUNCALV43	Variance of Uncalibrated Normal Visibility
QC DUNCALV63	Variance of Uncalibrated Normal Visibility
QC DUNCALV91	Variance of Uncalibrated Normal Visibility
QC FREQ MAX	Filter maximum frequency
QC FREQ MIN	Filter minimum frequency
QC LAMBDA117	Calibrated wavelength
QC LAMBDA127	Calibrated wavelength
QC LAMBDA232	Calibrated wavelength
QC LAMBDA43	Calibrated wavelength
QC LAMBDA63	Calibrated wavelength
QC LAMBDA91	Calibrated wavelength
QC PHOTA TARG1	Binned photometry beam A on Target
QC PHOTA TARG2	Binned photometry beam A on Target
QC PHOTA TARG3	Binned photometry beam A on Target
QC PHOTA TOTAL1	Binned photometry beam A on (Target+Sky)
QC PHOTA TOTAL2	Binned photometry beam A on (Target+Sky)
QC PHOTA TOTAL3	Binned photometry beam A on (Target+Sky)
QC PHOTB TARG1	Binned photometry beam B on Target
QC PHOTB TARG2	Binned photometry beam B on Target
QC PHOTB TARG3	Binned photometry beam B on Target
QC PHOTB TOTAL1	Binned photometry beam B on (Target+Sky)
QC PHOTB TOTAL2	Binned photometry beam B on (Target+Sky)
QC PHOTB TOTAL3	Binned photometry beam B on (Target+Sky)
QC SCN PROCESSED	Number of scans processed
QC SCN REJECTED	Number of scans rejected
QC T117	Measured Transfer Function
QC T127	Measured Transfer Function
QC T232	Measured Transfer Function
QC T43	Measured Transfer Function
QC T63	Measured Transfer Function
QC T91	Measured Transfer Function
QC TARFLUX1	Target Intensity beam station 1
QC TARFLUX2	Target Intensity beam station 2
QC TARSIZEX1	Target Width beam station 1
QC TARSIZEX2	Target Width beam station 2
QC TARSIZEY1	Target Height beam station 1
QC TARSIZEY2	Target Height beam station 2
QC TARX1	Target X beam station 1

QC TARX2	Target X beam station 2
QC TARY1	Target Y beam station 1
QC TARY2	Target Y beam station 2
QC THEOVIS	Theoretical Visibility
QC TRF BINNED1	Binned Transfer Function
QC TRF BINNED2	Binned Transfer Function
QC TRF BINNED3	Binned Transfer Function
QC TRF BINNED4	Binned Transfer Function
QC TRF BINNED5	Binned Transfer Function
QC UNCAL BINNED1	Uncalibrated Binned Visibility
QC UNCAL BINNED2	Uncalibrated Binned Visibility
QC UNCAL BINNED3	Uncalibrated Binned Visibility
QC UNCAL BINNED4	Uncalibrated Binned Visibility
QC UNCAL BINNED5	Uncalibrated Binned Visibility
QC UNCALV117	Uncalibrated Normal Visibility
QC UNCALV127	Uncalibrated Normal Visibility
QC UNCALV232	Uncalibrated Normal Visibility
QC UNCALV43	Uncalibrated Normal Visibility
QC UNCALV63	Uncalibrated Normal Visibility
QC UNCALV91	Uncalibrated Normal Visibility

C.5 QC parameter of the midi_acq recipe

QC Parameter	Short Description
QC ACQ TARFLUX1	Target Intensity beam station 1
QC ACQ TARFLUX2	Target Intensity beam station 2
QC ACQ TARSIZEX1	Target Width beam station 1
QC ACQ TARSIZEX2	Target Width beam station 2
QC ACQ TARSIZEY1	Target Height beam station 1
QC ACQ TARSIZEY2	Target Height beam station 2
QC ACQ TARX1	Target X beam station 1
QC ACQ TARX2	Target X beam station 2
QC ACQ TARY1	Target Y beam station 1
QC ACQ TARY2	Target Y beam station 2
QC GAUSS FIT X	Field of view x-center
QC GAUSS FIT Y	Field of view y-center
QC GAUSS FIT SIGMA X	Field of view x-width (from gaussfit)
QC GAUSS FIT SIGMA Y	Field of view y-center (from gaussfit)
QC GAUSS MEASURED FWHM AT X	Field of view FWHM (measured)
QC GAUSS MEASURED FWHM AT Y	Field of view FWHM (measured)
QC CENTROID X	Field of view x-center from flux barycenter
QC CENTROID Y	Field of view y-center from flux barycenter
QC CENTROID MEASURED FWHM AT X	Field of view FWHM (measured)
QC CENTROID MEASURED FWHM AT Y	Field of view FWHM (measured)

C.6 QC parameter of the midi_detlin recipe

QC Parameter	Short Description
QC DETLIN GR A0	zero order coeff
QC DETLIN GR A1	first order coeff
QC DETLIN GR A2	second order coeff
QC DETLIN GR A3	third order coeff
QC DETLIN GR NAME	Optical element
QC DETLIN GR SIG	Mean linearity Standart deviation
QC DETLIN IM A0	zero order coeff
QC DETLIN IM A1	first order coeff
QC DETLIN IM A2	second order coeff
QC DETLIN IM A3	third order coeff
QC DETLIN IM NAME	Optical element
QC DETLIN IM SIG	Mean linearity Standart deviation
QC DETLIN PR A0	zero order coeff
QC DETLIN PR A1	first order coeff
QC DETLIN PR A2	second order coeff
QC DETLIN PR A3	third order coeff
QC DETLIN PR NAME	Optical element
QC DETLIN PR SIG	Mean linearity Standart deviation

C.7 QC parameter of the midi_detron recipe

QC Parameter	Short Description
QC DETRON MEDIAN	Median of noise pattern

C.8 QC parameter of the midi_dsptrn recipe

QC Parameter	Short Description
QC TRN F1 GR CNT	number of count
QC TRN F1 GR TRANS	transmission
QC TRN F1 IM CNT	number of count
QC TRN F1 IM DX	size of the beam
QC TRN F1 IM DY	size of the beam
QC TRN F1 IM EXT	exposure time
QC TRN F1 IM FIL	Filter name for file1
QC TRN F1 IM NAME	dispersive element for file1
QC TRN F1 IM TRANS	transmission
QC TRN F1 IM X	Position of the beam
QC TRN F1 IM Y	Position of the beam
QC TRN F1 PR CNT	number of count

QC TRN F1 PR TRANS	transmission
QC TRN F2 GR CNT	number of count
QC TRN F2 GR TRANS	transmission
QC TRN F2 IM CNT	number of count
QC TRN F2 IM DX	size of the beam
QC TRN F2 IM DY	size of the beam
QC TRN F2 IM EXT	exposure time
QC TRN F2 IM FIL	Filter name for file1
QC TRN F2 IM NAME	dispersive element for file1
QC TRN F2 IM TRANS	transmission
QC TRN F2 IM X	Position of the beam
QC TRN F2 IM Y	Position of the beam
QC TRN F2 PR CNT	number of count
QC TRN F2 PR TRANS	transmission
QC TRN F3 GR CNT	number of count
QC TRN F3 GR TRANS	transmission
QC TRN F3 IM CNT	number of count
QC TRN F3 IM DX	size of the beam
QC TRN F3 IM DY	size of the beam
QC TRN F3 IM EXT	exposure time
QC TRN F3 IM FIL	Filter name for file1
QC TRN F3 IM NAME	dispersive element for file1
QC TRN F3 IM TRANS	transmission
QC TRN F3 IM X	Position of the beam
QC TRN F3 IM Y	Position of the beam
QC TRN F3 PR CNT	number of count
QC TRN F3 PR TRANS	transmission
QC TRN F4 GR CNT	number of count
QC TRN F4 GR TRANS	transmission
QC TRN F4 IM CNT	number of count
QC TRN F4 IM DX	size of the beam
QC TRN F4 IM DY	size of the beam
QC TRN F4 IM EX	exposure time
QC TRN F4 IM FIL	Filter name for file1
QC TRN F4 IM NAME	dispersive element for file1
QC TRN F4 IM TRANS	transmission
QC TRN F4 IM X	Position of the beam
QC TRN F4 IM Y	Position of the beam
QC TRN F4 PR CNT	number of count
QC TRN F4 PR TRANS	transmission
QC TRN F5 GR CNT	number of count
QC TRN F5 GR TRANS	transmission
QC TRN F5 IM CNT	number of count
QC TRN F5 IM DX	size of the beam
QC TRN F5 IM DY	size of the beam

QC TRN F5 IM EXT	exposure time
QC TRN F5 IM FIL	Filter name for file1
QC TRN F5 IM NAME	dispersive element for file1
QC TRN F5 IM TRANS	transmission
QC TRN F5 IM X	Position of the beam
QC TRN F5 IM Y	Position of the beam
QC TRN F5 PR CNT	number of count
QC TRN F5 PR TRANS	transmission
QC TRN F6 GR CNT	number of count
QC TRN F6 GR TRANS	transmission
QC TRN F6 IM CNT	number of count
QC TRN F6 IM DX	size of the beam
QC TRN F6 IM DY	size of the beam
QC TRN F6 IM EXT	exposure time
QC TRN F6 IM FIL	Filter name for file1
QC TRN F6 IM NAME	dispersive element for file1
QC TRN F6 IM TRANS	transmission
QC TRN F6 IM X	Position of the beam
QC TRN F6 IM Y	Position of the beam
QC TRN F6 PR CNT	number of count
QC TRN F6 PR TRANS	transmission

C.9 QC parameter of the midi_refpix recipe

QC Parameter	Short Description
QC PIX HS A1 ESI	window search size channel 1
QC PIX HS A1 EX	expected high_sensopen beam position channel 1
QC PIX HS A1 EY	expected high_sens beam position channel 1
QC PIX HS A1 SIZ	high_sens beam A size channel 1
QC PIX HS A1 X	high_sens beam A position channel 1
QC PIX HS A1 Y	high_sens beam A position channel 1
QC PIX HS A2 ESI	window search size channel 2
QC PIX HS A2 EX	expected high_sensopen beam position channel 2
QC PIX HS A2 EY	expected high_sens beam position channel 2
QC PIX HS A2 SIZ	high_sens beam size channel 2
QC PIX HS A2 X	high_sens beam position channel 2
QC PIX HS A2 Y	high_sens beam position channel 2
QC PIX HS B1 ESI	window search size channel 1
QC PIX HS B1 EX	expected high_sensopen beam position channel 1
QC PIX HS B1 EY	expected high_sens beam position channel 1
QC PIX HS B1 SIZ	high_sens beam A size channel 1
QC PIX HS B1 X	high_sens beam A position channel 1
QC PIX HS B1 Y	high_sens beam A position channel 1
QC PIX HS B2 ESI	window search size channel 2

QC PIX HS B2 EX	expected high_sensopen beam position channel 2
QC PIX HS B2 EY	expected high_sens beam position channel 2
QC PIX HS B2 SIZ	high_sens beam size channel 2
QC PIX HS B2 X	high_sens beam position channel 2
QC PIX HS B2 Y	high_sens beam position channel 2
QC PIX OP A ESI	window search size channel 1
QC PIX OP A EX	expected open beam position channel 1
QC PIX OP A EY	expected open beam position channel 1
QC PIX OP A SIZ	open beam size channel 1
QC PIX OP A X	open beam position channel 1
QC PIX OP A Y	open beam position channel 1
QC PIX OP B ESI	window search size channel 2
QC PIX OP B EX	expected open beam position channel 2
QC PIX OP B EY	expected open beam position channel 2
QC PIX OP B SIZ	open beam size channel 1
QC PIX OP B X	open beam position channel 1
QC PIX OP B Y	open beam position channel 1
QC PIX SP A1 ESI	window search size channel 1
QC PIX SP A1 EX	expected high_sensopen beam position channel 1
QC PIX SP A1 EY	expected high_sens beam position channel 1
QC PIX SP A1 SIZ	high_sens beam A size channel 1
QC PIX SP A1 X	high_sens beam A position channel 1
QC PIX SP A1 Y	high_sens beam A position channel 1
QC PIX SP A2 ESI	window search size channel 2
QC PIX SP A2 EX	expected high_sensopen beam position channel 2
QC PIX SP A2 EY	expected high_sens beam position channel 2
QC PIX SP A2 SIZ	high_sens beam size channel 2
QC PIX SP A2 X	high_sens beam position channel 2
QC PIX SP A2 Y	high_sens beam position channel 2
QC PIX SP B1 ESI	window search size channel 1
QC PIX SP B1 EX	expected high_sensopen beam position channel 1
QC PIX SP B1 EY	expected high_sens beam position channel 1
QC PIX SP B1 SIZ	high_sens beam A size channel 1
QC PIX SP B1 X	high_sens beam A position channel 1
QC PIX SP B1 Y	high_sens beam A position channel 1
QC PIX SP B2 ESI	window search size channel 2
QC PIX SP B2 EX	expected high_sensopen beam position channel 2
QC PIX SP B2 EY	expected high_sens beam position channel 2
QC PIX SP B2 SIZ	high_sens beam size channel 2
QC PIX SP B2 X	high_sens beam position channel 2
QC PIX SP B2 Y	high_sens beam position channel 2
QC PIX SP PA ESI	window search size photometric beam
QC PIX SP PA EX	expected high_sens beam position photometric beam
QC PIX SP PA EY	expected high_sens beam position photometric beam
QC PIX SP PA SIZ	high_sens beam size photometric beam

QC PIX SP PA X	high_sens beam position photometric beam
QC PIX SP PA Y	high_sens beam position photometric beam
QC PIX SP PB ESI	window search size photometric beam B
QC PIX SP PB EX	expected high_sens beam position photometric beam B
QC PIX SP PB EY	expected high_sens beam position photometric beam B
QC PIX SP PB SIZ	high_sens beam size photometric beam B
QC PIX SP PB X	high_sens beam position photometric beam B
QC PIX SP PB Y	high_sens beam position photometric beam B

C.10 QC parameter of the midi_wavecal recipe

QC Parameter	Short Description
QC WCAL CH117 R1	wavelength calibration
QC WCAL CH117 R2	wavelength calibration
QC WCAL CH117 R3	wavelength calibration
QC WCAL CH117 R4	wavelength calibration
QC WCAL CH127 R1	wavelength calibration
QC WCAL CH127 R2	wavelength calibration
QC WCAL CH127 R3	wavelength calibration
QC WCAL CH127 R4	wavelength calibration
QC WCAL CH232 R1	wavelength calibration
QC WCAL CH232 R2	wavelength calibration
QC WCAL CH232 R3	wavelength calibration
QC WCAL CH232 R4	wavelength calibration
QC WCAL CH43 R1	wavelength calibration
QC WCAL CH43 R2	wavelength calibration
QC WCAL CH43 R3	wavelength calibration
QC WCAL CH43 R4	wavelength calibration
QC WCAL CH63 R1	wavelength calibration
QC WCAL CH63 R2	wavelength calibration
QC WCAL CH63 R3	wavelength calibration
QC WCAL CH63 R4	wavelength calibration
QC WCAL CH91 R1	wavelength calibration
QC WCAL CH91 R2	wavelength calibration
QC WCAL CH91 R3	wavelength calibration
QC WCAL CH91 R4	wavelength calibration
QC WCAL VAR R1	wavelength calibration
QC WCAL VAR R2	wavelength calibration
QC WCAL VAR R3	wavelength calibration
QC WCAL VAR R4	wavelength calibration

D Troubleshooting Guide

<TBD>