

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

GIRAFFE Pipeline User Manual

VLT-MAN-ESO-19500-3883

Issue 15

Date 2020-01-21

Prepared:	GIRAFFE Pipeline Team	2020-01-21	
	Name	Date	Signature
Approved:	P.Ballester		
	Name	Date	Signature
Released:	M. Peron		
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
0.1	2006-03-01	All	First version
1.0	2006-03-17	All	Version for public release
2.5	2007-12-28	All	Updated to pipeline version 2.5
2.5.1	2008-02-25	All	QC parameters updated
2.8.6	2010-03-01	All	Updated to pipeline version 2.8
2.8.7	2010-08-06	All	New bias correction method added
2.9	2012-05-11	All	Updated to pipeline version 2.9
2.10	2013-03-15	All	Updated to pipeline version 2.10
6/2.11	2013-04-30	All	Updated to pipeline version 2.11
7/2.13	2014-07-04	9.6.4, 9.6.5	Added parameters for SDP format
8/2.14	2015-02-17	All	Updated to pipeline version 2.14
9/2.14.2	2015-02-17	9.5.2, 9.6.2	Improved parameter description of the flat.apply option
10/2.14.3	2016-02-15	All	Updated to pipeline version 2.14.3
11/2.15	2016-07-08	All	Software version updated to 2.15
12/2.16	2017-02-03	All	Software version updated to 2.16
13/2.16.3	2018-03-15	1.1	Typo fixed
		A.1, A.2.1	Cleaned up list of supported platforms and requirements
14/2.16.4	2019-02-22	1	Inaccessible URLs updated.
15	2020-01-21	Change record	Do not use package version as issue revision
		1.1, 1.3	Put document scope related information into the appropriate section
		A.1, A.2.1, A.2.3	Cleaned up supported platforms, requirements, and build instructions

This page was intentionally left blank

Contents

1 Introduction **9**

1.1 Purpose . . . 9

1.2 Acknowledgements . . . 9

1.3 Scope . . . 9

1.4 Applicable documents . . . 9

1.5 Reference documents . . . 10

1.6 Notational conventions . . . 10

1.7 Abbreviations and acronyms . . . 10

2 Overview **11**

3 GIRAFFE Instrument Description **12**

3.1 Instrument overview . . . 12

3.2 GIRAFFE instrument capabilities . . . 12

4 Quick start **16**

4.1 GIRAFFE pipeline recipes . . . 16

4.2 An introduction to Gasgano and EsoRex . . . 16

4.2.1 Using Gasgano . . . 17

4.2.2 Using EsoRex . . . 23

5 Known Problems **26**

6 Instrument Data Description **27**

6.1 Data format description . . . 27

6.2 Raw frames . . . 28

7 Static Calibration Data **32**

7.1 Grating data table . . . 32

7.2 ThAr line catalog . . . 33

7.3 Slit geometry tables . . . 33

7.4 Line masks . . . 34

7.5 Atmospheric extiction table . . . 35
7.6 Spectrophotometric Standards table . . . 35
7.7 Other calibration data . . . 36

8 Data Reduction **37**
8.1 Data reduction overview . . . 37
8.2 Calibration cascade . . . 37

9 Pipeline Recipes Interfaces **40**
9.1 gimasterbias . . . 40
9.1.1 Input frames . . . 41
9.1.2 Recipe parameters . . . 41
9.1.3 Detailed parameter description . . . 41
9.1.4 Extra recipe parameters . . . 42
9.1.5 Product frames . . . 42
9.1.6 Quality control parameters . . . 43
9.2 gimasterdark . . . 43
9.2.1 Input frames . . . 43
9.2.2 Recipe parameters . . . 43
9.2.3 Detailed parameter description . . . 44
9.2.4 Extra recipe parameters . . . 44
9.2.5 Product frames . . . 45
9.2.6 Quality control parameters . . . 45
9.3 gimasterflat . . . 46
9.3.1 Input frames . . . 47
9.3.2 Recipe parameters . . . 47
9.3.3 Detailed parameter description . . . 49
9.3.4 Extra recipe parameters . . . 51
9.3.5 Product frames . . . 54
9.3.6 Quality control parameters . . . 55
9.4 giwavecalibration . . . 56

9.4.1 Input frames . . . 56
9.4.2 Recipe parameters . . . 59
9.4.3 Detailed parameter description . . . 59
9.4.4 Extra recipe parameters . . . 60
9.4.5 Product frames . . . 64
9.4.6 Quality control parameters . . . 65
9.5 gistandard . . . 66
9.5.1 Input frames . . . 66
9.5.2 Recipe parameters . . . 67
9.5.3 Detailed parameter description . . . 68
9.5.4 Extra recipe parameters . . . 69
9.5.5 Product frames . . . 70
9.5.6 Quality control parameters . . . 72
9.6 giscience . . . 72
9.6.1 Input frames . . . 73
9.6.2 Recipe parameters . . . 74
9.6.3 Detailed parameter description . . . 75
9.6.4 Extra recipe parameters . . . 76
9.6.5 Product frames . . . 78
9.6.6 Quality control parameters . . . 80

10 Algorithms **81**
10.1 General Algorithms . . . 81
10.1.1 Bias removal . . . 81
10.1.2 Dark subtraction . . . 82
10.1.3 Fibre localisation . . . 82
10.1.4 Spectrum extraction . . . 83
10.1.5 Wavelength calibration . . . 84
10.2 Recipe algorithms . . . 84
10.2.1 gimasterbias . . . 84
10.2.2 gimasterdark . . . 85

10.2.3 gimasterflat . . . 85
10.2.4 giwavecalibration . . . 85
10.2.5 gistandard . . . 86
10.2.6 giscience . . . 86

A Installation **88**

A.1 Supported platforms . . . 88
A.2 Building the GIRAFFE pipeline . . . 89
A.2.1 Requirements . . . 89
A.2.2 Downloading the GIRAFFE pipeline distribution . . . 89
A.2.3 Compiling and installing the GIRAFFE pipeline . . . 89
A.2.4 Configuring the pipeline recipe front–end applications . . . 90

B Quality control parameter definitions **91**

1 Introduction

1.1 Purpose

The GIRAFFE pipeline is a subsystem of the *VLT Data Flow System* (DFS). It is used in two operational environements, for the ESO *Data Flow Operations* (DFO), 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 GIRAFFE pipeline recipes are made public to the user community, to allow a more personalised processing of the data from the instrument. The purpose of this document is to describe a typical GIRAFFE data reduction sequence with the GIRAFFE pipeline.

1.2 Acknowledgements

The GIRAFFE pipeline is based on the GIRAFFE Base Line Data Reduction Software (BLDRS) developed by the Observatoire de Genève (OGL). This software package is, at the time of writing this document, available on the web at `http://girbldrs.sourceforge.net`. We would like to thank the BLDRS software team, André Blecha, Gilles Simond and Frederic Royer for their excellent work done on the BLDRS software package and their contributions and support.

1.3 Scope

This document describes the GIRAFFE pipeline used at ESO-Garching and ESO-Paranal for the purpose of data assessment and data quality control. It is a complete description of the data reduction recipes implemented by the GIRAFFE pipeline, reflecting the status of the GIRAFFE pipeline as of April, 1st 2016 and it applies to all releases of the GIRAFFE pipeline 2.17 release series.

The examples on running individual pipeline recipes in this manual use the *EsoRex* command and manually created list of input files. Several interfaces to automatically organise the data, create the list of input files and execute the pipeline recipes in the proper sequence are available, see the ESO Pipeline web page *https://www.eso.org/pipelines* for details.

Please note that the use of Gasgano as a GUI for processing data is deprecated. Its use is no longer recommended and the related section in this manual, as well as support for Gasgano as a data processing GUI application in general will be dropped entirely in a future release.

1.4 Applicable documents

[AD1]	VLT Data Flow System Specifications for Pipeline and Quality Control	VLT–SPE–ESO–19600–1233
[AD2]	Data Flow for VLT Instruments Requirement Specification	VLT–SPE–ESO–19000–1618/2.0
[AD3]	DFS Pipeline & Quality Control — User Manual	VLT–MAN–ESO–19500-1619
[AD4]	ESO DICB — Data Interface Control Document	GEN–SPE–ESO–19400–0794
[AD5]	Common Pipeline Library User Manual	VLT–MAN–ESO–19500–2720
[AD6]	FLAMES Calibration Plan	VLT–PLA–ESO–13700–3248

1.5 Reference documents

[RD1]	FLAMES User Manual	VLT–MAN–ESO–13700–2994
[RD2]	FLAMES–UVES Data Reduction Users' Manual	VLT–MAN–ESO–19500–3016
[RD3]	Gasgano User's Manual	VLT–PRO–ESO-19000–1932
[RD4]	EsoRexUser Manual	VLT–MAN–ESO–19500–xxxx
[RD5]	De Cuyper J.–P., Hensberge H., 1998, A&AS 128, 409	
[RD6]	Horne K. 1986, PASP 98, 609	
[RD7]	FITS format description for pipeline products with data, error and data quality information	VLT–MAN–ESO–19500–5667
[RD8]	3D Visualization Tool Manual	VLT–MAN–ESO–19500–5651

1.6 Notational conventions

Hierarchical FITS keyword names, appearing in the document, are given using the dot–notation to improve readability. This means, that the prefix "HIERARCH ESO" is left out, and the spaces separating the keyword name constituents in the actual FITS header are replaced by a single dot.

1.7 Abbreviations and acronyms

BLDRS	Base Line Data Reduction Software
CPL	Common Pipeline Library
DFS	Data Flow System
DFSD	Data Flow Systems Departement
DMD	Data Management and Operations Division
DFO	Data Flow Operations
DO	Data Organiser
DRS	Data Reduction System
ESO	European Southern Observatory
FITS	Flexible Image Transport System
FLAMES	Fibre Large Array Multi Element Spectrograph
FPOSS	Fibre Positioner Obervation Support Software
OGL	Observatiore de Genève
OS	Observing software
PAF	VLT parameter file format
QC	Quality Control
RB	Reduction Block
RBS	Reduction Block Scheduler
SOF	Set Of Frames
VLT	Very Large Telescope

2 Overview

In collaboration with instrument consortia, the Data Flow Systems Department (DFSD) 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.*, combined bias frames, super-flats, wavelength dispersion solutions).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes (*e.g.*, combined ISAAC jitter stacks; bias-corrected, flat-fielded FORS images, wavelength-calibrated UVES spectra). 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 the command line, from the automatic data management tools available on Paranal or from graphical user interfaces.

ESO offers two front-end applications for launching pipeline recipes, Gasgano and EsoRex, both included in the pipeline distribution (see appendix A, page 88). These applications can also be downloaded separately from `http://www.eso.org/gasgano` and `http://www.eso.org/cpl/esorex.html`. An illustrated introduction to Gasgano is provided in the "Quick Start" section of this manual (see page 16).

The GIRAFFE instrument and the different types of GIRAFFE raw frames and auxilliary data are described in sections 3, 6, and 7. In addition, a description of the GIRAFFE data frame types, and the pipeline processing scheme used by DFO, is also available at `http://www.eso.org/qc/GIRAFFE/pipeline/pipe_gen.html`.

A brief introduction to the usage of the available reduction recipes using Gasgano or EsoRex is presented in section 4. In section 5 we advice the user about known data reduction problems providing also possible solutions.

An overview of the data reduction, what are the input data, and the recipes involved in the calibration cascade is provided in section 8.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in section 9.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in section 10.

The installation of the GIRAFFE pipeline is described in appendix A and the list of definitions of the implemented quality control parameters can be found in appendix B.

3 GIRAFFE Instrument Description

3.1 Instrument overview

The GIRAFFE[1] instrument is part of the Fibre Large Array Multi Element Spectrograph (FLAMES) of the VLT. Mounted on the Nasmyth A focus of UT2, FLAMES can access targets over a field of view of 25 arcmin diameter. The FLAMES facility consist of 3 main subsystems coordinated by a special Observing Software (FLAMES OS) and 2 auxiliary subsystems. The three main subsystems are:

OzPoz: The fibre positioner, hosting to plates. While one plate is observing, the fibres are positioned on the second plate, in preparation for a following observation, by a robotic arm. Figure 3.2 shows a configured fibre positioner plate.

GIRAFFE: A medium-high resolution optical spectrograph, GIRAFFE, with three types of feeding fibre systems: MEDUSA, IFU, ARGUS.

UVES Fiber link: 8 fibres feeding the UVES spectrograph.

The GIRAFFE spectrograph and fibre system was developed by the Observatoire de Paris-Meudon (P.I. F. Hammer) and Observatoire de Genève (P.I. A. Blecha), in tight collaboration with ESO. ESO–Garching is responsible for the instrument control electronics hardware and software, the cryogenic system and the detector system (FIERA). The GIRAFFE integration started in ESO–Garching in spring 2001. The general layout of the GIRAFFE spectrograph is shown in figure 3.1 and figure 3.3 shows GIRAFFE on the Nasmyth platform of UT2.

A complete documentation of FLAMES in general, and the GIRAFFE spectrograph in particular, is available in the FLAMES User Manual [RD1] which is available at `http://www.eso.org/sci/facilities/paranal/instruments/flames/overview.html`.

This manual describes only the pipeline recipes which are used to process data obtained from the GIRAFFE spectrograph. For information in the FLAMES–UVES pipeline please refer to [RD2].

3.2 GIRAFFE instrument capabilities

GIRAFFE is a medium-high ($R = 5600 - 46000$) resolution spectrograph for the entire visible range, from 370 nm to 950 nm. It is equipped with two gratings and several filters are available to select the required spectral range. Five additional fibres allow simultaneous wavelength calibration of every exposure. Each object can be only observed in one, or a fraction of a single echelle order at once. The fibre system feeding GIRAFFE consists of the following components

MEDUSA: The MEDUSA fibres are single fibres, which allow up to 132 separate objects (including sky fibres) to be observed in a single observation. Two separate sets of MEDUSA fibres exists, one per positioner plate. Each fibre has an aperture of 1.2 arcsec on the sky.

[1] The name GIRAFFE originates from the first design, where the spectrograph was placed vertically.

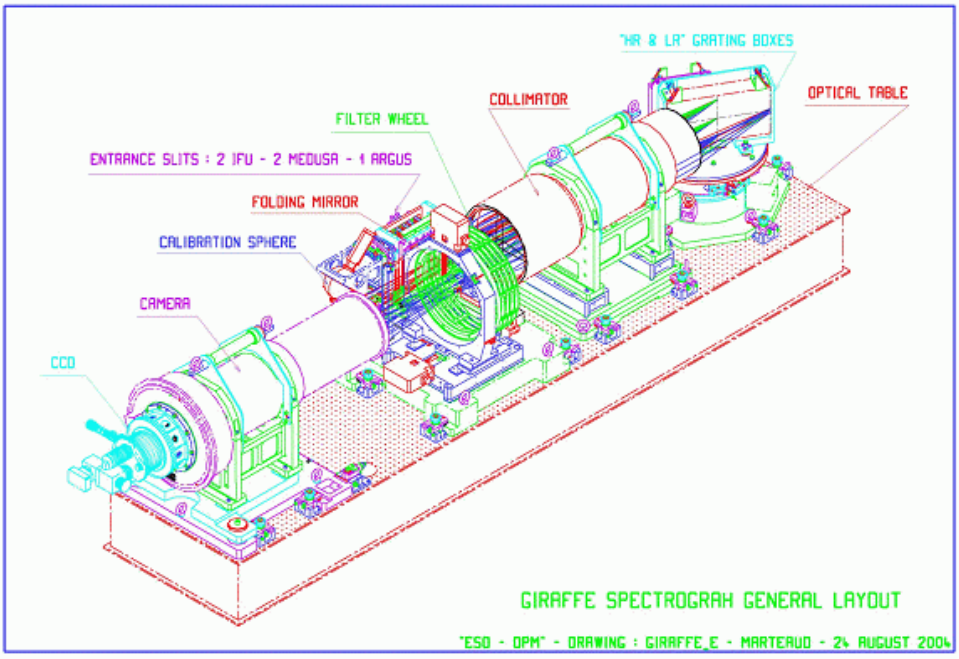


Figure 3.1: The general layout of the GIRAFFE spectrograph.

IFU: The deployable Integral Field Units consists of a rectangular array of 20 microlenses of 0.52 arcsec each, giving an aperture of 2 arcsec $\times$ 3 arcsec. For each plate there are 15 IFU units dedicated to objects and another 15 dedicated to sky measurements. In the latter, only the central fibre is present.

ARGUS: This is a large integral field unit, which is mounted at the centre of one plate of the fibre positioner and consists of a rectangular array of 22 $\times$ 14 microlenses. Two magnification scales are available: 1 : 1 with a sampling of 0.52 arcsec per microlens and a total aperture of 11.5 arcsec $\times$ 7.3 arcsec, and 1:1.67 with 0.3 arcsec per microlens and a total aperture of 6.6 arcsec $\times$ 4.2 arcsec. In addition, 15 ARGUS sky fibres can be positioned in the 25 arcmin field.

GIRAFFE is equipped with one 2 K $\times$ 4 K EEV CCD (15 μm pixels), with a scale of 0.3 arcsec pixel^{-1} in MEDUSA, IFU and ARGUS direct mode, and a scale of 0.15 arcsec pixel^{-1} in the enlarged ARGUS mode. GIRAFFE is operated with 39 fixed setups (31 high resolution + 8 low resolution modes), available for each of the 5 different fibre slit systems.



Figure 3.2: One of the two plates of the fibre positioner OzPoz. The fibres are attached to the plate using magnetic buttons.

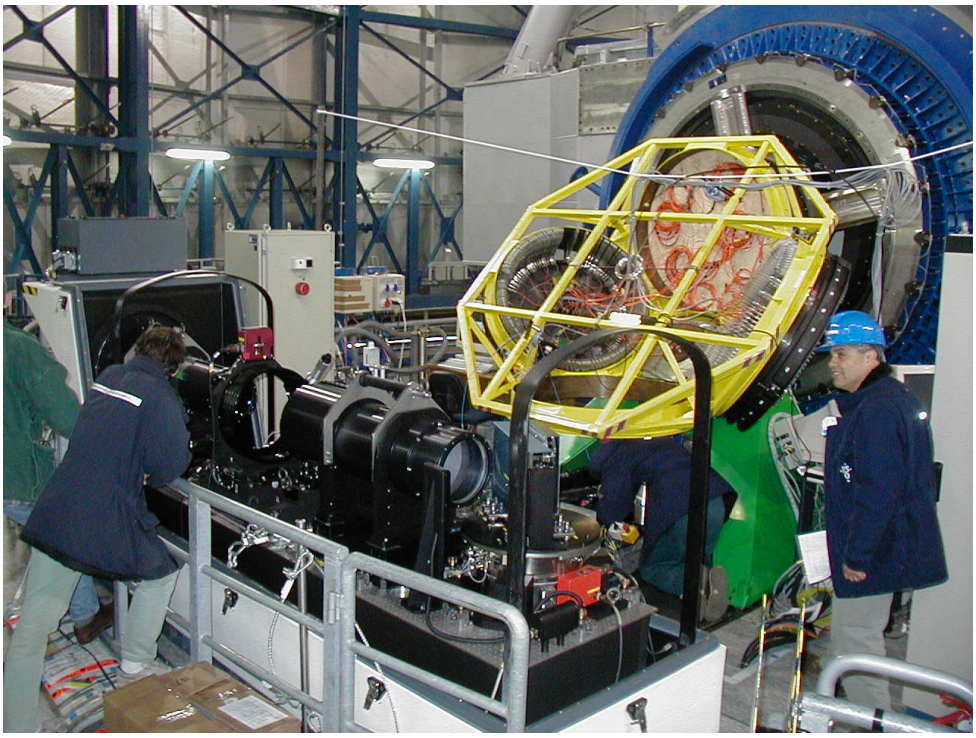


Figure 3.3: The GIRAFFE spectrograph on the Nasmyth platform of UT2. Behind the GIRAFFE spectrograph the fibre positioner OzPoz is visible.

4 Quick start

This section outlines the GIRAFFE pipeline recipes, and how they can be used from the two available applications EsoRexand Gasgano, to reduce GIRAFFE observations.

4.1 GIRAFFE pipeline recipes

For processing GIRAFFE raw and calibration data of all instrument setups, the GIRAFFE pipeline provides, with the current pipeline release, 5 data reduction recipes, 2 technical recipes used to monitor the detector performance, and 1 utility recipe to do a simple combination of input frames. Here only the 5 data reduction recipes are described, since the 3 other recipes are not used in the normal workflow for calibration and science data.

gimasterbias generates a master bias frame from a set of raw bias frames, and, optionally a bad pixel map of the CCD detector.

gimasterdark generates a master dark frame from a set of raw dark frames.

gimasterflat determines the position and width of the spectra on the CCD from a set of flat field lamp images. In addition the recipe creates a product containing the extracted flat field lamp spectra, and a model of the scattered light contamination.

giwavecalibration determines a complete dispersion solution from a Thorium–Argon arc lamp image. The created dispersion solution consists of an instrument physical model and additional correction terms.

gistandard creates fully reduced standard star observations by applying the calibrations produced by the three other calibration recipes, and computes the instrument response and the efficiency curves. The recipe is only applicable to the *IFU* and *Argus* instrument modes.

giscience: creates fully reduced scientific observations by applying the calibrations produced by the three other recipes. For GIRAFFE IFU observations, i.e. for the instrument modes *IFU* and *Argus* a reconstructed image of the field of view(s) is also created.

4.2 An introduction to Gasgano and EsoRex

In order to use the GIRAFFE pipeline recipes for processing any set of GIRAFFE observational data, these data must be properly classified, and the appropriate calibration data must be associated to them.

Data Classification means that an identifier is assigned to a data frame, which describes its contents in a unique way. This identifier is then used by the pipeline recipes to decide in which way a particular data frame is used during processing. *Data Classification* is based on the FITS keywords found in the header of each data frame and usually the three FITS keywords DPR.CATG, DPR.TYPE, and DPR.TECH are used for this purpose, although additional keywords may also be used. The identifier of the data type assigned to a data frame is called *DO Category*.

Data Association is the process of selecting appropriate calibration data for the reduction of a set of raw data frames. *Data Association* is also based on the information stored in the FITS header of the data frames. For

the reduction of one or more raw data frames, the required calibration data is selected from a repository using a set of FITS keywords, called *association keywords*, which is specific for each type of calibration frame. *Association keywords* are typically keywords, which uniquely identify a particular instrument setup. Using these keywords calibration data whose *association keywords* match their counterparts in the FITS header of the raw data frame are selected from the calibration data repository (for instance the calibration database) as suitable for the processing the raw frame.

Data Classification and *Data Association* are the two parts of a process known as *Data Organisation*.

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, from the automatic data management tools available at Paranal, or from the graphical application Gasgano.

Gasgano is a data management tool that simplifies the data organisation process, offering automatic data classification and making the data association easier (*although an automatic association of frames 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.

4.2.1 Using Gasgano

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

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

```
gasgano
```

Figure 4.1 shows the Gasgano main window. Directories containing the GIRAFFE raw and calibration data files can be added using the pull–down menu *File->Add/Remove Files*. The files located in the selected directories are loaded, classified, and their names appear in the upper part of the Gasgano main window, with the *DO Category* shown in the column *Classification*.

The hierarchical view of the data files may be customised by the user, by defining FITS keywords in the preferences dialog, accessible through the menu item *File->Preferences*, which are then used to group the data files. An obvious grouping scheme for GIRAFFE observations is a grouping with respect to the slit system and grating setup used, which can be realised using the keywords INS.SLIT.NAME and INS.EXP.MODE for grouping[2]. An example is shown in Figure 4.2.

A left click on a file name in the data browser panel will display the frame's FITS header in the lower part of the Gasgano main window. This view allows also to search the header for a given string or to apply a filter to it. A right click on one of the displayed file names opens a context menu which allows to apply one of the listed actions to the selected file, for instance displaying it with the users preferred viewer (customisable in the *Preferences* dialog).

[2]The keyword INS.EXP.MODE is only present in data taken after October 2004. For older data the keywords INS.GRAT.ID, INS.GRAT.WLEN and INS.GRAT.ORDER may be used instead.

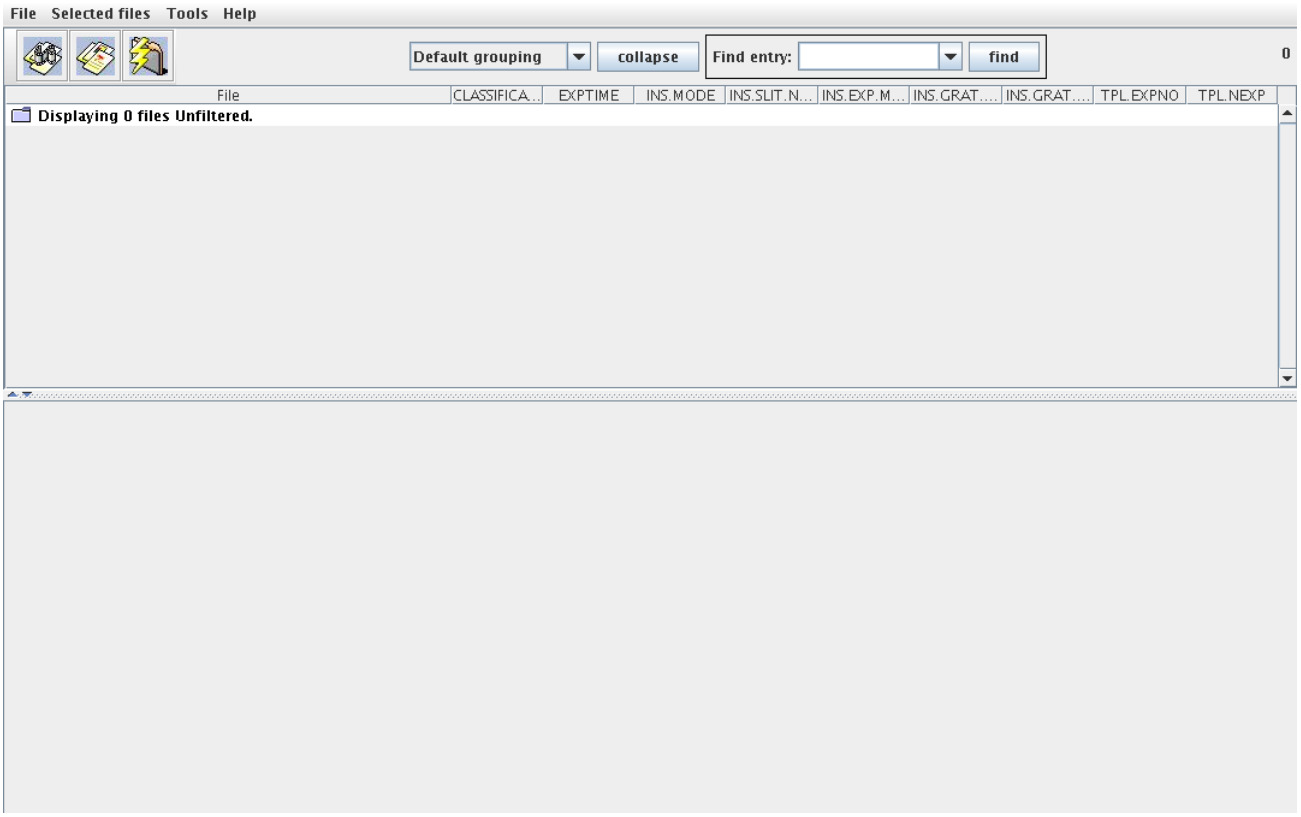


Figure 4.1: The Gasgano main window as it appears immediately after starting up. No files have been yet, therefore the data browser panel is still empty.

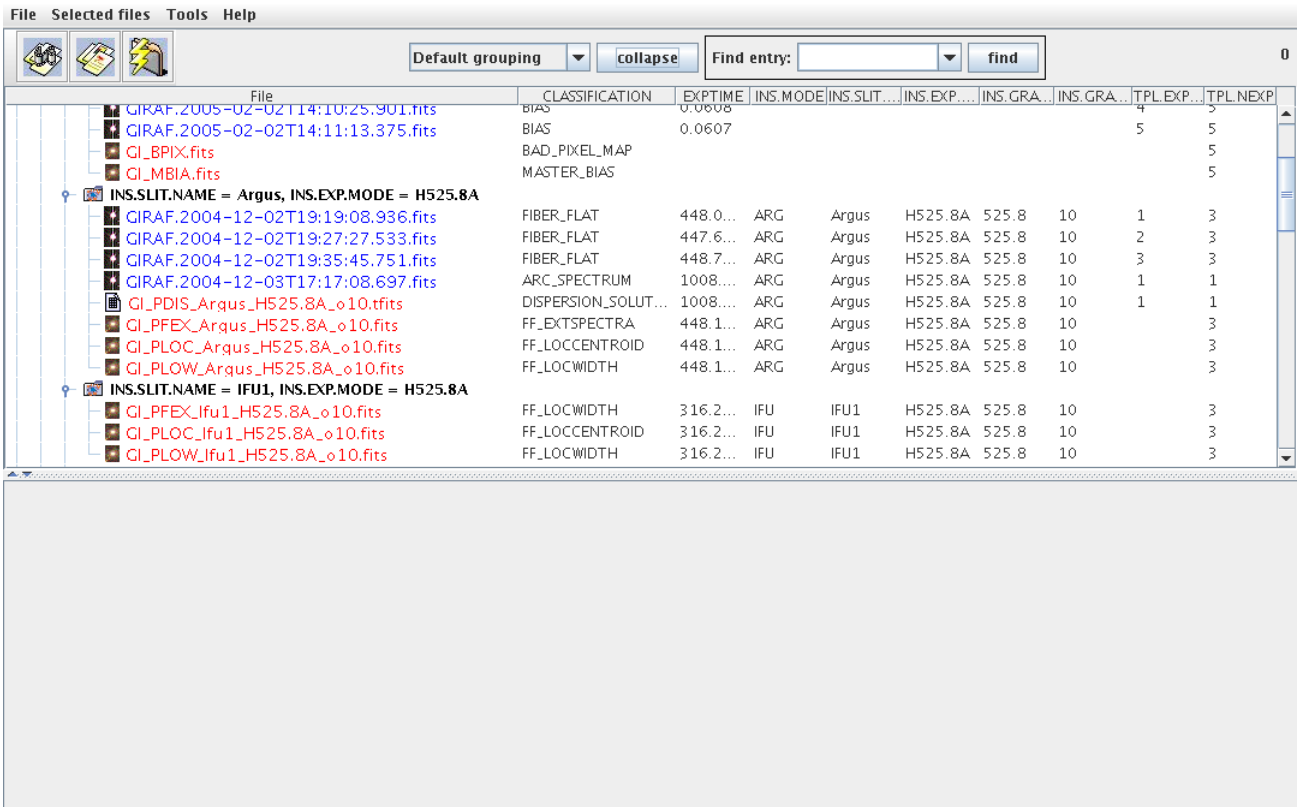


Figure 4.2: The data browser panel of the Gasgano main window shows a set of GIRAFFE raw and calibration files, which have been grouped using the keywords INS.SLIT.NAME and INS.EXP.MODE.

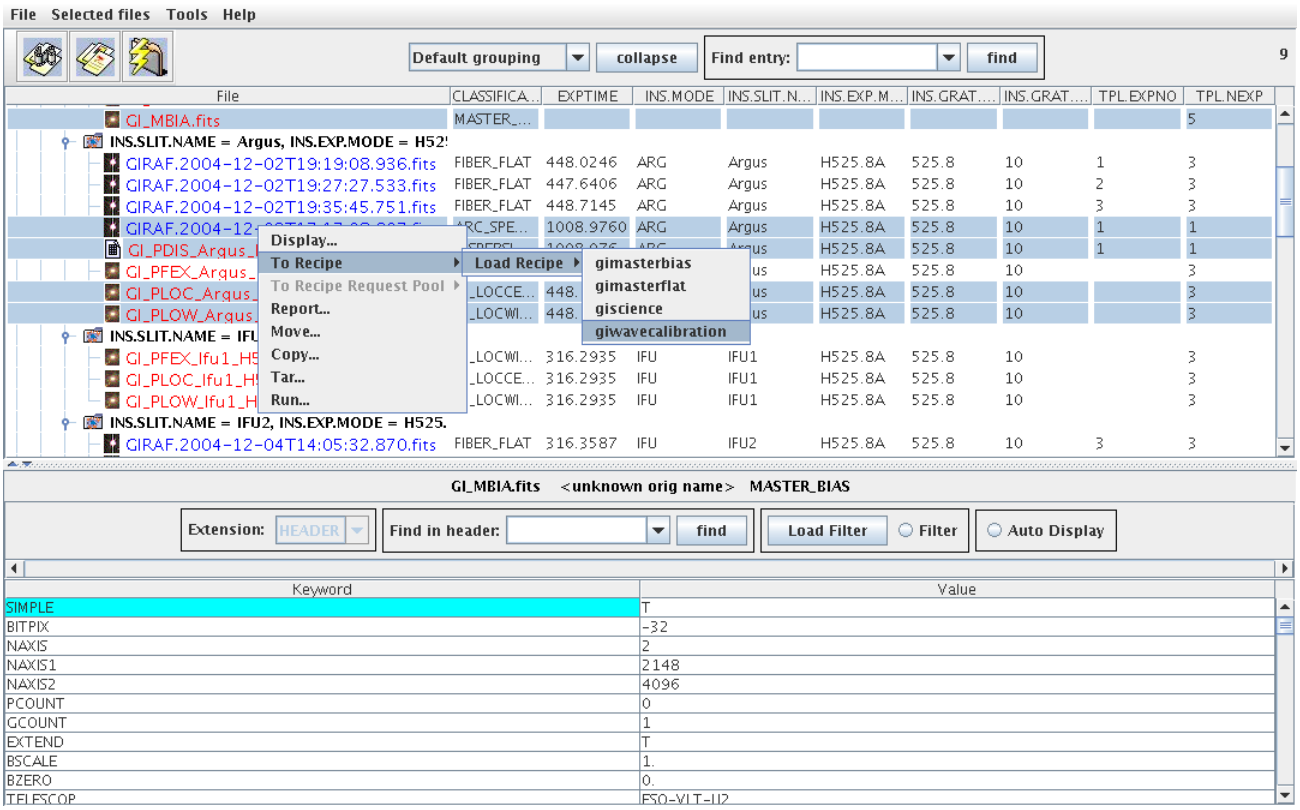


Figure 4.3: A set of raw and calibration data frames have been selected in the data browser panel and are to be sent to the GIRAFFE recipe *giwavecalibration*.

Frames to be processed by one of the GIRAFFE pipeline recipes are selected in the data browser panel. The selected set of raw and calibration frames is then sent to the desired recipe using the *To Recipe* menu item from the context menu, which opens on right clicking into the data browser panel, or from the pull–down menu *Selected files->To Recipe*. On selecting the recipe, the recipe execution window opens, showing the previously selected frames listed in the *Input Frames* panel. An example is shown in Figure 4.3.

A short on–line help for each recipe is available from the *Help* menu of the recipe execution window. The list of available recipe parameters is shown in the *Paramters* panel of the recipe execution window together with their default and current value (see figure 4.4. The parameters may be changed here as needed and the final recipe configuration can be saved, for re–using it later, by selecting the *Save Current Settings* menu item from the *File* menu of the recipe execution window.

After the recipe parameters have been customised the recipe is executed by a left click on the *Execute* button. The text output of the recipe is displayed in the *Log Messages* panel at the bottom of the recipe execution window as the recipe runs. The recipe output can also be saved after the recipe stopped. On successful completion of the recipe, the created product frames are listed in the *Output Frames* panel, showing also their *DO Category*, and the buttons to display the frames and to locate them in the, updated, data browser view of the main window. An example of a successful recipe execution is shown in figure 4.5.

For a complete overview of the Gasgano application and a detailed description on how to use its graphical

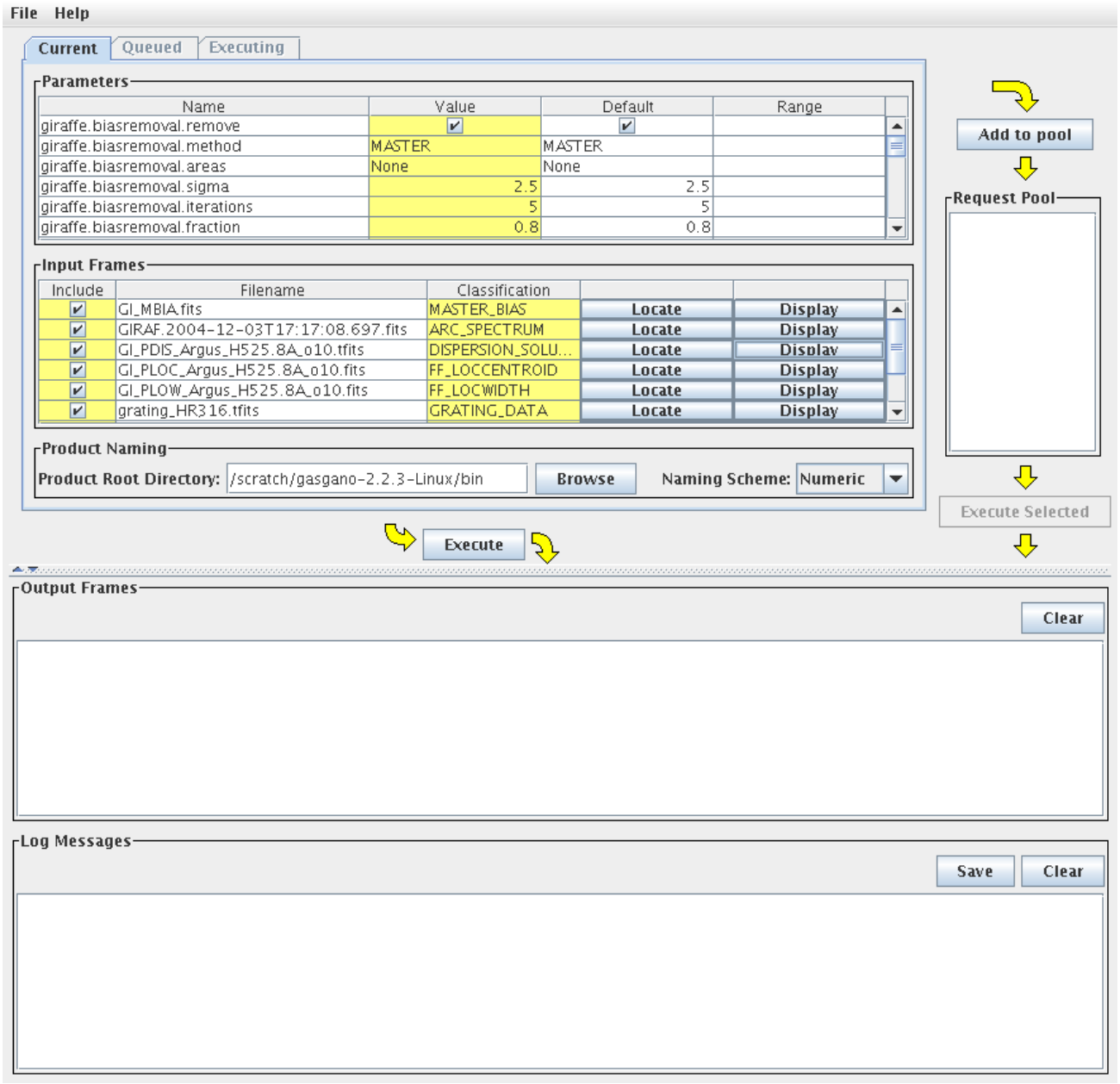


Figure 4.4: The recipe execution window, showing the loaded GIRAFFE recipe *giwavecalibration*, the list of available recipe parameters and the set of raw and calibration frames.

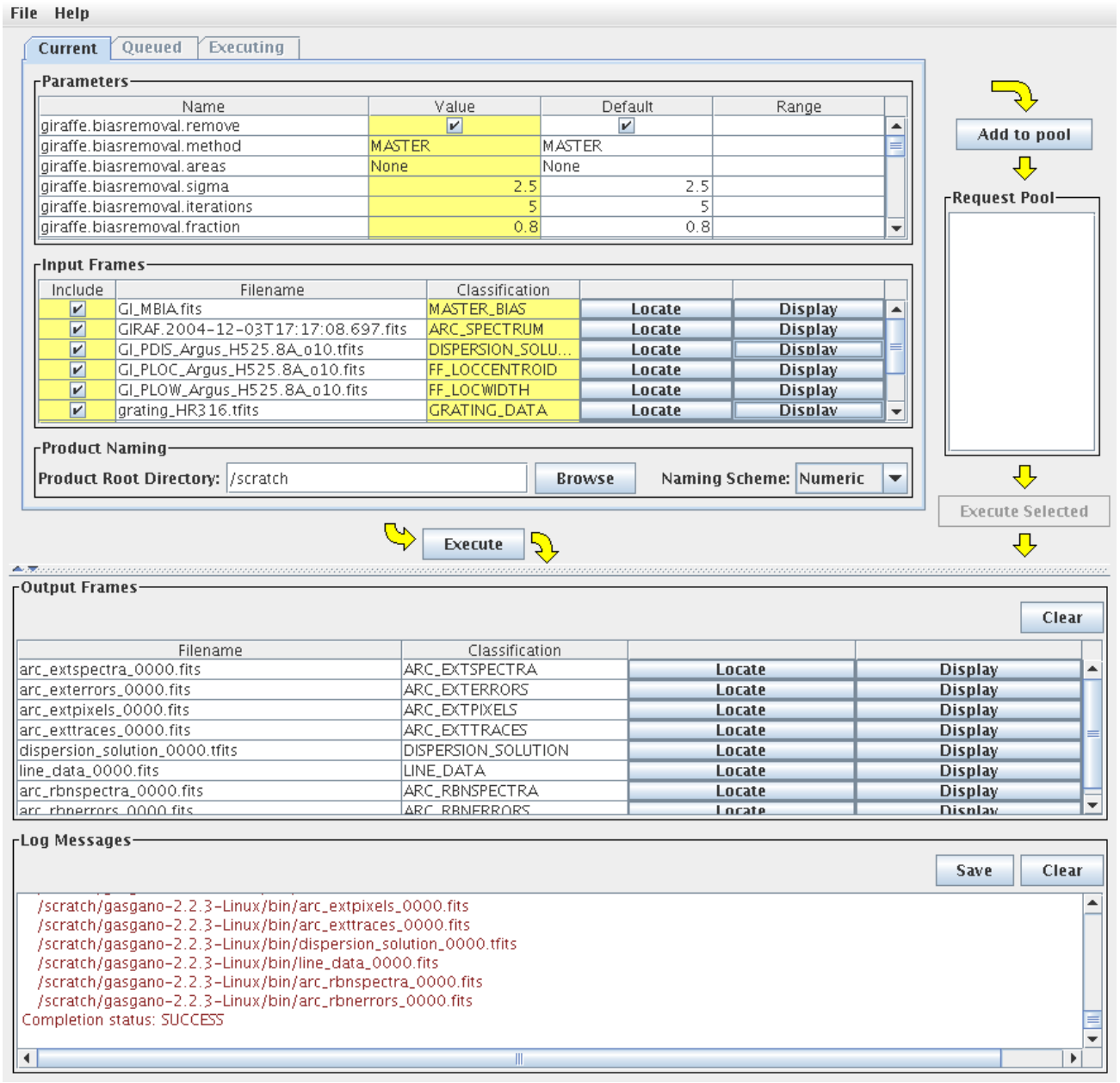


Figure 4.5: On successful completion, the *Output Frames* panel is updated with the created recipe products.

interface, please refer to the *Gasgano User's Manual* [RD3].

4.2.2 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 (see Section 6.2 on page 28). 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:

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* (SOF)[3], which is required when launching a recipe.

The following shows an example SOF, as it could be used for the wavelength calibration recipe *giwavecalibration*:

```
GIRAF.2005-04-17T22:59:03.001.fits ARC_SPECTRUM
/diskc/giraffe/cal/GI_MBIA.fits MASTER_BIAS
/diskc/giraffe/cal/GI_PLOC_Ifu1_H572.8_o10.fits FF_LOCCENTROID
/diskc/giraffe/cal/GI_PLOW_Ifu1_H572.8_o10.fits FF_LOCWIDTH
/diskc/giraffe/cal/GI_PDIS_Ifu1_H572.8_o10.fits DISPERSION_SOLUTION
/diskc/giraffe/cal/slit_geometry_ifu1_H572.8_o10.fits SLIT_GEOMETRY_SETUP
/diskc/giraffe/cal/line_catalog_ThAr.fits LINE_CATALOG
/diskc/giraffe/cal/grating_HR316.fits GRATING_DATA
```

The syntax for each line is the path to the input file, followed by the file's *DO Category*.

The GIRAFFE pipeline recipes do not cross–check in any way that the given *DO Category* corresponds to the actual contents of the associated input file specified in the SOF. The reason for this is, that the recipes are just one component of a complete pipeline installation running on Paranal, and therefore assume that data classification and association are done beforehand. 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 4.2.1).

This feature may even be used. Since the format of the raw GIRAFFE images is fixed by the fibre pseudo slit in the focal plane, regardless of the actual light source being observed, i.e. calibration lamp or scientific target, the pipeline recipes would not notice any difference between a science or calibration exposure. Therefore any type of raw frame using the fibre system could in principle be used as input frame for the recipe *giscience*. For instance, using an arc lamp frame as input to the recipe *giscience* allows a quick assessment of the quality of the dispersion solution. But, in general, this feature should only be used with the necessary care.

[3]The set-of-frames corresponds to the *Input Frames* panel of the Gasgano recipe execution window (see Figure 4.4).

In general, feeding a pipeline recipe with an SOF that contains improper *DO categories* for the corresponding input frames, the recipe will, at a certain point stop with a more or less appropriate error message, but there might be cases where the recipe completes without any apparent indication that the created data products are invalid because of using invalid input data.

EsoRex syntax:

For a detailed documentation on how to configure *EsoRex* please refer to [RD4]. 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 customise *EsoRex* run the command:

esorex - -help

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 listed by the command

esorex - -params recipe_name

A short description of each parameter accepted by a given recipe is shown by executing:

esorex - -help recipe_name

To get description of a particular recipe the command

esorex - -man-page recipe_name

must be executed.

Recipe configuration:

For each pipeline recipe an *EsoRex* configuration file may be created , which can be used to store the preferred parameter settings for a recipe[4] Parameter values specified in a recipe configuration file override the recipe's built–in default values, and can, themselves, be overridden by an command line option. The configuration files are usually stored in the directory `$HOME/.esorex`, and have the same name as the recipe for which they have been created, with the file name extension '.rc' appended.

The following example creates an *EsoRex* configuration file for the recipe *gimasterbias*, which will be called 'gimasterbias.rc'.

esorex - -create-config gimasterbias - -stack-method=median

The generated configuration file will contain a line

[4] An *EsoRex* recipe configuration file corresponds to the column labled *Default* in the *Parameters* panel of the Gasgano recipe execution window (see Figure 4.4).

```
giraffe.stacking.method=median
```

where 'giraffe.stacking.method' is the fully qualified parameter name corresponding to the command line alias 'stack-method'. This line changes the default value of the bias frame combination method from its built–in value 'average' to 'median' for subsequent recipe runs, as long as the option 'stack-method' is not given on the command line. When the configuration file is created, *EsoRex* will add comments which allow to associate the full parameter names to their corresponding command line option aliases.

The built–in parameter default values of a recipe *recipe_name* may be saved to a file calling esorex like this:

esorex - -create-config recipe_name

An *EsoRex* configuration file, which is not located in the directory $HOME/.esorex, or has a name other than 'recipe_name.rc' must be explicitly specified on the *EsoRex* command line:

esorex - -recipe-config=path_to_configuration_file

Recipe execution:

A GIRAFFE pipeline recipe can be executed using *EsoRex* by passing the recipe name, recipe parameters (optional), and a set–of–frames file to it. The command line given as an example below, would run the GIRAFFE science reduction recipe using the input frames specified in the SOF file 'science.sof' using the recipes default settings:

esorex giscience science.sof

If recipe parameters should be passed as command line options, they have to be given immediately after the recipe name. Parameters given on the command line override any parameter default either built–in or defined in a configuration file. Command line options given before the recipe name are interpreted as options to *EsoRex* itself.

EsoRex examples: The following lines show a few *EsoRex* command line examples for running GIRAFFE pipeline recipes.

```
esorex --output-prefix=test gimasterbias gimasterbias.sof
esorex gimasterflat --fiber-nspec=132 ../medusa1/H572.8/gimasterflat.sof
esorex giwavecalibration --wcal-lswidth=10,10,10 giwavecalibration.sof
```

The list of parameters for each GIRAFFE pipeline recipe is given in section 9 together with a short description of their purpose.

A complete description on how to use *EsoRex*, is available at http://www.eso.org/cpl/esorex.html and the EsoRexUser Manual [RD4].

5 Known Problems

The list of problems, already known at the time of a new release of the GIRAFFE pipeline is always available in the file BUGS, which is part of the GIRAFFE pipeline distribution.

Currently the GIRAFFE pipeline does not handle well the following situations:

1. The bluest instrument setups are difficult to calibrate due to large gradients and differences in contrast which are present over the wavelength range of these setups.
2. Bad columns may confuse the fibre detection algorithm. This is most likely the case if the bad column just “touches” one of the nearby spectra.
3. The treatment of incomplete spectra visible at the right edge of the CCD needs to be improved. Currently it is recommended that the fibre localisation is checked after the flat–field recipe has finished.
4. The optimal extraction method should currently be used only for MEDUSA observations. For Argus and IFU data the extraction algorithm is very slow.
5. For Argus and IFU the fitting of the fibre profile has to be improved. Cross-talk between adjacent fibers has an impact on the estimated profile parameters. Since these profile parameters are used by the Horne and optimal extraction method, artefacts can be present in the extracted spectra.

These issues will be addressed in a future release.

6 Instrument Data Description

GIRAFFE data can be separated into *raw* frames and *product* frames. Raw frames are the unprocessed output of the GIRAFFE instrument, while product frames are the result of the GIRAFFE pipeline processing. In addition to that the GIRAFFE pipeline uses a set of (static) calibration files, like line catalog, or grating data.

Any raw or product frame can be classified on the basis of a set of keywords read from its header. Usually the data is classified by a dedicated application, the *Data Organiser* (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 master bias frame) can be obtained by matching the values of a different set of header keywords.

Each kind of raw frame is typically associated to a single GIRAFFE 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.

In the following all supported raw and product GIRAFFE data frames are listed, together with the keywords used for their classification and correct association. The given *DO category* is the identifier which is expected by the GIRAFFE pipeline recipes for that kind of data frame. This is the identifier which has to be used in the set–of–frames file to allow the recipes to find their required input data frames (c.f. section 4.2.2 on page 23).

6.1 Data format description

Raw GIRAFFE FITS files, as they are generated by the instrument, are FITS files which have the image data stored in the FITS primary data unit. In case an obvservation has been done using one of the three fibre systems, two FITS binary table extensions are also present. The first extension, the OzPoz table contains the list of the magnetic buttons placed on one of the fibre positioner plates. The contents of this table is variable, since it includes the association of the objects and the fibre buttons. It contains also the fibre positioner keywords. The OzPoz table is essentially the PAF file generated by FPOSS (the fibre positioner observation support software tool), with some OzPoz information added (for instance R, θ and errors).

The second table extension is the FLAMES FIBRE table. It contains the fibre description and the association between fibre buttons and fibre position in the slits and on the detector, and it also includes lab-measured fibre transmission values. This table is static.

Apart from a few exceptions GIRAFFE pipeline product images are stored using an extracted format, i.e. only the extracted data from fibres, which were actually allocated, are stored in the pixel columns of the product images. The relation between the pixel column index and the fibre's position in the slit or on the detector is given in a housekeeping table, attached to the products as a FITS binary table extension. This table is called FIBER_SETUP and contains essentially the merged information of the OzPoz table and the FLAMES FIBRE table present in the raw data frame. The important columns for associating an object spectrum with the data stored in the pixel columns are the columns named INDEX, FPS, and FPD. The column INDEX is the spectrum number as it has been detected by the pipeline, and it is also the index of the pixel column of the product image used to store the data values. The columns FPS and FPD are the fibre position within the slit and on the detector, respectively.

While the FIBRE_SETUP table will always contain the columns SSN, PSSN, RP, Retractor, R, THETA, ORIENT, TYPE, RA, and DEC, there are columns which are only present depending on the contents of the (dynamic) OzPoz table and also on the instrument mode used. The columns X and Y, which give the coordinates

Column	Origin	Comments
INDEX	pipeline	Pixel column index of spectral data in the product image
FPS	fibre table	Fibre position in the slit
SSN	fibre table	Subslit number the fibre belongs to
PSSN	fibre table	Fibre position within the subslit
RP	fibre table	Retractor position on the plate. This number corresponds to the fibre number used e.g. in FPOSS.
X	fibre table	X position of the fibre in the IFU head and the reconstructed image matrix. (Only present for IFU and Argus observations)
Y	fibre table	Y position of the fibre in the IFU head and the reconstructed image matrix. (Only present for IFU and Argus observations)
Retractor	fibre table	Serial number of the retractor
FPD	fibre table	Fibre position on the detector; for Medusa and IFU identical to FPS, for Argus the order is reversed.
OBJECT	OzPoz table	Object identifier (from Target Setup File, column 1)
R	OzPoz table	Button R (polar coordinate) position [μm]
THETA	OzPoz table	Button θ position on plate [rad]
ORIENT	OzPoz table	Button orientation.
TYPE	OzPoz table	Object Type.
RA	OzPoz table	Right Ascension (from Target Setup File, column 2)
DEC	OzPoz table	Declination (from Target Setup File, column 3)
MAGNITUDE	OzPoz table	Target Magnitude (from Target Setup File, column 6; science data only)

Table 6.1: Overview of the columns which may be present in the FIBER_SETUP table which is present in almost all pipeline product frames.

of the fibres in the IFU heads will only be present, if the observation was using the IFU or Argus fibre systems. Table 6.1 gives an overview of the contents of the FIBER_SETUP table.

In addition, the FIBER_SETUP table is complemented by some pipeline recipes with computed extra information, for instance the fibre transmission and the barycentric correction.

A detailed description of the GIRAFFE raw data binary tables can be found in the FLAMES User Manual [RD1], and is also available at `http://www.eso.org/observing/dfo/quality/GIRAFFE/pipeline/bin_tables.html`. The latter contains also the description of the fibre setup table used for the pipeline product images.

6.2 Raw frames

Although the GIRAFFE spectrograph is fed by different fibre systems, which could be interpreted as different observing modes, there is no difference in the spectral format or the reduction process of the data coming from the Medusa, IFU and Argus fibres. Therefore the different fibre systems are not used in the GIRAFFE classification rules, and are not reflected the GIRAFFE *DO categories*. It is of course taken into account for the association of calibration data. As a consequence, a science observation is always classified as `SCIENCE`, regardless of which fibre system was used.

In the following the GIRAFFE raw frames, which can be processed by the current pipeline version, are listed together with the recipe to be used to process each frame type, the keywords used for association and classification and the assigned *DO category*.

Bias:

Recipe: `gimasterbias`

Classification:

DO Category	DPR.CATG	DPR.TYPE	DPR.TECH
BIAS	CALIB	BIAS	IMAGE

Association keywords:
INSTRUME
DET.CHIP1.ID
DET.WIN1.BINX
DET.WIN1.BINY
DET.READ.MODE
DET.READ.SPEED
DET.READ.CLOCK

Dark:

Recipe: `gimasterdark`

Classification:

DO Category	DPR.CATG	DPR.TYPE	DPR.TECH
DARK	CALIB	DARK	IMAGE

Association keywords:
INSTRUME
DET.CHIP1.ID
DET.WIN1.BINX
DET.WIN1.BINY
DET.READ.MODE
DET.READ.SPEED
DET.READ.CLOCK

Flat field: Standard robotic flat field exposure or Nasmyth screen flat field exposure.

Recipe: `gimasterflat`

Classification:

DO Category	DPR.CATG	DPR.TYPE	DPR.TECH
FIBER_FLAT	CALIB	LAMP,FLAT	MOS
FIBER_FLAT	CALIB	LAMP,FLAT,NASMYTH	MOS
FIBER_FLAT	CALIB	LAMP,FLAT	IFU
FIBER_FLAT	CALIB	LAMP,FLAT,NASMYTH	IFU

Association keywords:
INSTRUME
INS.GRAT.ID
ESO.INS.SLIT.NAME
ESO.INS.EXP.MODE
DET.CHIP1.ID
DET.WIN1.BINX
DET.WIN1.BINY
DET.READ.MODE
DET.READ.SPEED
DET.READ.CLOCK

Arc lamp exposure:

Recipe: `giwavecalibration`

Classification:

DO Category	DPR.CATG	DPR.TYPE	DPR.TECH
ARC_SPECTRUM	CALIB	LAMP,WAVE	MOS
ARC_SPECTRUM	CALIB	LAMP,WAVE	IFU

Association keywords:
INSTRUME
INS.GRAT.ID
ESO.INS.SLIT.NAME
ESO.INS.EXP.MODE
DET.CHIP1.ID
DET.WIN1.BINX
DET.WIN1.BINY
DET.READ.MODE
DET.READ.SPEED
DET.READ.CLOCK

Standard star exposure: Spectrophotometric standard star observations using the Argus or IFU instrument modes.

Recipe: `gistandard`

Classification:

DO Category	DPR.CATG	DPR.TYPE	DPR.TECH
STD	CALIB	STD,SimCal	IFU
STD	CALIB	STD,OzPoz	IFU

Association keywords:
INSTRUME
INS.GRAT.ID
INS.SLIT.NAME
INS.EXP.MODE
DET.CHIP1.ID
DET.WIN1.BINX
DET.WIN1.BINY
DET.READ.MODE
DET.READ.SPEED
DET.READ.CLOCK

Science exposure: Science frames with and without simultaneous calibration fibres are supported

Recipe: `giscience`

Classification:

DO Category	DPR.CATG	DPR.TYPE	DPR.TECH
SCIENCE	SCIENCE	OBJECT,SimCal	MOS
SCIENCE	SCIENCE	OBJECT,OzPoz	MOS
SCIENCE	SCIENCE	OBJECT,SimCal	IFU
SCIENCE	SCIENCE	OBJECT,OzPoz	IFU

Association keywords:
INSTRUME
INS.GRAT.ID
INS.SLIT.NAME
INS.EXP.MODE
DET.CHIP1.ID
DET.WIN1.BINX
DET.WIN1.BINY
DET.READ.MODE
DET.READ.SPEED
DET.READ.CLOCK

7 Static Calibration Data

This section describes the static calibration tables, which are used by the GIRAFFE pipeline. Contrary to calibration products, which are created by the calibration recipes, the static calibration data has been prepared manually. These tables contain usually data derived from catalogs, or from the design of the instrument.

The *DO Category* to be used for any of the static calibration data files is given by the value of the FITS header keyword `PRO.CATG`.

7.1 Grating data table

The grating tables, one for each of the two available gratings, contain the description of the grating in terms of physical parameters. In addition, average parameter values for the GIRAFFE optical model for each instrument setup are present. These model parameters allow for bootstrapping the GIRAFFE pipeline, in case standard solutions are not provided, or are not suitable. Table 7.1 gives an overview of the information present in the grating tables.

Column	Description
SETUP	Unique identifier of the instrument setup.
ETC	Name of the setup, as it is used by the ETC.
ORDER	Grating diffraction order
WLMIN	Minimum wavelength of the setup [nm]
WLEN0	Central wavelength of the setup [nm]
WLMAX	Maximum wavelength of the setup [nm]
BAND	Band pass of the setup [nm]
RIFA	Resolution for the IFU and Argus fibre systems
RMED	Resolution for the Medusa fibre system
FCOLL	Collimator focal length [mm]
GCAM	Camera magnification factor
SDX	Slit offset in the dispersion direction [mm]
SDY	Slit offset in the spatial direction [mm]
SPHI	Slit rotation angle [rad]
THETA	Grating angle [rad]

Table 7.1: The contents of the grating data tables.

The grating tables are stored in the files `grating_HR316.fits` and `grating_LR600.fits` containing the data for the high resolution and low resolution grating respectively.

The *DO category* for grating tables is `GRATING_DATA` and the association keywords are `INSTRUME` and `INS.GRAT.ID`.

Column	Description
WLEN	Line wavelength [nm]
NAME	Name of the emitting ion
FLUX	Line flux in arbitrary units
COMMENT	Comments to indicate saturated or blended lines

Table 7.2: The contents of the ThAr line catalog.

7.2 ThAr line catalog

The Thorium–Argon line catalog used by the GIRAFFE pipeline is based on the catalog of De Cuyper and Hensberge [RD5], and has been provided by OGL. Usually it is not necessary to edit this line catalog, since the GIRAFFE pipeline provides a sophisticated line selection algorithm, which works well for all possible instrument setups. But if it should be necessary to, permanently, remove a line from the catalog, it is sufficient to edit the comment field in the catalog. Any comment longer than 3 characters instructs the pipeline to ignore this line during the selection process. Table 7.2 shows the structure of the line catalog.

The line catalog is stored in the file `line_catalog.fits`.

The *DO category* for the line catalog is `LINE_CATALOG` and the association keyword is `INSTRUME`.

7.3 Slit geometry tables

The slit geometry gives the the position of each fibre (X_f, Y_f) in the focal plane. It is used, for instance, to predict the position of a spectral line on the detector, by using the GIRAFFE optical model. The slit geometry comes in two flavours, as a generic template for each slit system, and a calibrated version for each combination of slit system and instrument setup. Table 7.3 show the structure of both types of slit geometry tables.

The setup specific slit geometry tables can be generated from the templates and are bound to a particular fibre setup, i.e. they are valid only for the fibre localisation for which they have been generated. But even the setup specific slit geometries can be considered as 'static', since experience has shown that the slit geometries do not need to be updated regularly, provided that the fibre localisation is not changed[5].

The GIRAFFE calibration database contains a pre–generated slit geometry for each possible setup, which can be used in conjunction with the also provided standard fibre localisations.

The slit geometry template files are called `slit_geometry_`*`slitname`*`.fits`, where *slitname* is one of: `medusa1`, `medusa2`, `ifu1`, `ifu2`, or `argus`. The names of the setup specific slit geometry tables are named like `slit_geometry_`*`slitname_setup`*`nm_o`*`order`*`.fits`. *setup* and *order* have to be replaced with the identifier of the grating and the central wavelength of an instrument setup, for instance H484.5, and the grating order of this setup.

The *DO category* to be used for the slit geometry templates tables is `SLIT_GEOMETRY_MASTER`. For the

[5]Even this is only an issue if a the fibre localisation changed in such a way that it contains more fibres than an existing slit geometry table. If a fibre localistion contains less fibres than the slit geometry table, the pipeline uses only the position information of the fibres of the given localisation.

Column	Template	Setup	Description
INDEX	X	X	Running number
FPS	X	X	Fibre position in the slit
STATUS	X		Unused
SSN	X	X	Sub–slit number
XF	X	X	Fibre position in the focal plane (dispersion direction) [mm]
YF	X	X	Fibre position in the focal plane (spatial direction) [mm]
ZF	X	X	Unused
ZDEFOCUS	X	X	Unused
PSSN	X		Fibre position in the sub–slit
RV		X	Residual fibre offset in terms of radial velocity [km/s]
RVERR		X	Uncertainty of the residual fibre offset [km/s]
RESOLUTION		X	Resolution derived from the fitted width of the correlation peak

Table 7.3: The contents of the template and setup specific slit geometry tables. The presence of the individual columns in the two types of slit geometries is indicated.

Column	Description
WLEN1	Start wavelength of the interval [nm]
WLEN2	End wavelength of the interval [nm]
DEEP	Unused
WIDTH	Width of the wavelength interval [nm]
WLENCENTER	Wavelength of the interval center [nm]

Table 7.4: The contents of the line mask table.

setup specific slit geometry tables the *DO category* is `SLIT_GEOMETRY_SETUP`.

The association keywords for the template are `INSTRUME` and `INS.SLIT.NAME`. For the setup specific slit geometries the keyword `INS.EXP.MODE` has to be used in addition.

7.4 Line masks

Line masks are needed for the creation of the setup specific slit geometry tables. They define a list of wavelength intervals, which are centered on calibration lines from the line catalog. The mask, generated from this list of intervals, is cross–correlated with a rebinned arc–lamp spectrum to detemine residual position offsets for each fibre.

The line mask files are called like `line_mask_ThAr_`*`slitname_setup`*`nm_o`*`order`*`.fits`, where *setup* and *order* have to be replaced with the grating identifier and the central wavelength of an instrument setup, and the corresponding grating order.

The *DO category* for line mask tables is `LINE_MASK`, and the keywords to be used for the association are `INSTRUME` and `INS.EXP.MODE`.

Column	Description
LAMBDA	Wavelength [nm]
LA_SILLA	Extinction coefficient [mag/airmass]

Table 7.5: The contents of the atmospheric extinction table.

Column	Description
OBJECT	Object identifier
TYPE	Spectral type
VMAG	Visual magnitude [mag]
RA_2000	Right ascension J2000 [H:M:S]
DEC_2000	Declination J2000 [H:M:S]
RA_DEG	Right ascension J2000 [°]
DEC_DEG	Declination J2000 [°]
NDATA	Number of valid data points in the data vectors
LAMBDA	Central wavelength of the spectral bin [Å]
BIN_WIDTH	Width of the spectral bin [Å]
F_LAMBDA	Fluxes of the spectral bin [$\mathrm{ergs^{-1}cm^{-2}\AA^{-1}}$]

Table 7.6: The contents of the spectrophotometric standards table.

7.5 Atmospheric extiction table

For the computation of the instrument response the correction of the atmospheric extinction is required. The atmospheric extinction table contains, for the given wavelength, the site specific extinction coefficients. Currently only data for La Silla is available.

The atmospheric extinction table is called `atmospheric_extinction.fits`.

The *DO category* to be used for the atmospheric extinction table is ATMOSPHERIC_EXTINCTION, and the association keyword is INSTRUME.

7.6 Spectrophotometric Standards table

The spectrophotometric standards table is required for the computation of the instrument response and contains the reference spectra for all spectrophotometric standard star which are routinely observed with GIRAFFE. The table, and its data is basically identical to the one used for UVES, from which it was inherited[6]. The table contains one row for each spectrophotometric standard star, where the wavelength, the size of the wavelength bin and the corresponding fluxes are stored in arrays.

The spectrophotometric standards table is called `flux_standards.fits`.

[6]The storage format of this table may however be changed into a more straight forward layout in the future.

The *DO category* to be used for the spectrophotometric standards table is FLUX_STANDARDS, and the association keyword is INSTRUME.

7.7 Other calibration data

Apart from these static calibration data standard calibration solutions are available, as part of the GIRAFFE pipeline kit, from the ESO web pages (http://www.eso.org/sci/software/pipelines). These standard calibration solutions may be used as reference input for the various recipes. This is the recommended way to start building calibration solutions from one's own GIRAFFE observations.

These calibration products, which have been created with the pipeline described in this document, are provided for convenience of the user. But they can in principle be recreated from scratch using the provided pipeline recipes, if it should turn out to be necessary. However, one should take into account that rebuilding calibrations from scratch may be a time consuming task.

In particular the pipeline recipes *gimasterflat* and *giwavecalibration* (cf. 9) can reuse an already existing version of their products, the fibre localisation and the dispersion solution, respectively, and update or adjust them to match the actually provided raw data frames.

This set of standard solutions therefore contains for each supported instrument setup the necessary:

- fibre localisation centroid positions,
- fibre width data,
- extracted flat-field lamp spectra,
- dispersion solution.
- setup specific slit-geometry tables

Details about these pipeline calibration products, and how to use them can be found in section 9.

8 Data Reduction

This section gives an overview on how GIRAFFE observations can be reduced using the GIRAFFE pipeline recipes.

8.1 Data reduction overview

The sequence of data reduction steps to be carried out is pretty much straight forward for a fibre coupled spectrograph:

- Remove the bias.
- Remove the dark.
- Locate the fibres on the detector.
- Extract the spectra.
- Apply flat–field and fibre–to–fibre transmission correction.
- Apply the dispersion solution to establish a wavelength scale.
- Reconstruct an image of the FOV(s) in case of IFU or Argus observations.

The GIRAFFE science reduction recipe *giscience* fine tunes the wavelength scale by default, if the simultaneous calibration lamps were used during the exposure. It also computes the barycentric, heliocentric and geocentric corrections for the spectra, if all necessary information is found in the raw data frames. However the corrections are not applied to the reduced spectra, but added to the data product as extra information.

The recipe *giscience* also does not provide a sky subtration, since it is assumed that the selection of the sky spectra to use requires a decision that can only be taken with the appropriate scientific background knowledge. Instead the recipe provides the reduced science observations with all corrections applied to both, object and sky spectra, so that the sky subtraction can easily be carried out using the preferred standard data reduction package.

The required calibration products can either be generated, using the GIRAFFE calibration recipes, obtained from the data packages shipped by the Data Flow Operations department, or from the set of standard calibrations provided as part of the pipeline kit.

The set of provided standard calibrations is complete, so that in principle running just the pipeline recipe *giscience* on a scientific observation gives a first, quick–look result.

8.2 Calibration cascade

The Calibration cascade is a schematic representation of the dependencies of recipes on raw frames, calibration products and static calibration data. An illustration of the GIRAFFE calibration cascade is the association map shown in Figure 8.1.

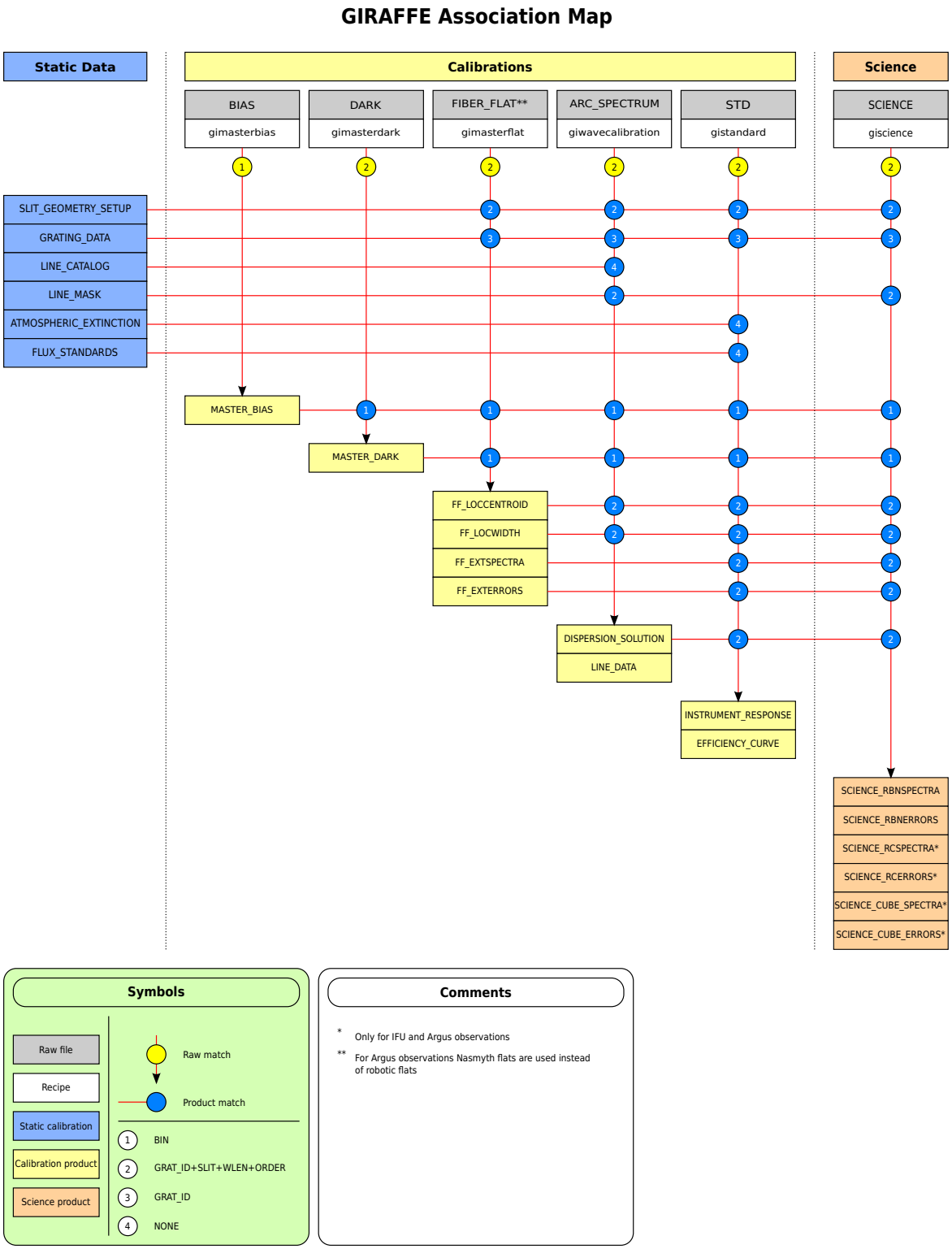


Figure 8.1: The GIRAFFE Association Map. This shows the GIRAFFE calibration cascade as it is implemented by DFO.

To create the required calibration products for the reduction of science observations the pipeline recipes *gimasterbias*, *gimasterdark*, *gimasterflat* and *giwavecalibration* must be run in that order. As an optional last step the pipeline recipe *gistandard* recipe may be used for Argus and IFU observations to obtain an estimate of the instrument response. The respective raw frames, calibration products and static calibration data needed by each recipe are summarised in section 9.

The onset of the GIRAFFE calibration cascade is the recipe *gimasterbias*, which creates a master bias frame from a set of raw bias frames, and optionally a map of bad detector pixels. The created bad pixel map is an optional input to subsequent recipes and is used during the spectrum extraction task.

The second recipe to run is *gimasterdark*. With modern CCD detectors the removal of the dark is usually not strictly necessery. But, in the case of GIRAFFE observations carried out before May 15th, 2008 with the CCD *Bruce*, the removal of the dark may be needed to remove the prominent glow feature in the upper right part of the CCD. It shape and extension is seen best in the dark frames. Observations taken after the detector *Bruce* has been replaced by the new CCD *Carreras* there is no such feature anymore, and the dark correction can safely be skipped.

The next recipe to execute is the recipe *gimasterflat*. Although its name suggests that the main product of the recipe is a master flat–field, this is only a by–product. It actual task is to detect and trace the fibres on the master flat–field, which is created from a set of raw flat fields. The recipe records the location and width of the detected spectra in two calibration products, the localisation centroid frame and the localisation width frame. Using the fibre positions and widths an extracted and normalised flat–field is also created.

The dispersion solution is computed from a ThAr arc lamp calibration frame using the recipe *giwavecalibration*. Starting from the fibre positions obtained from the recipe *gimasterflat*, the arc lamp spectra are extracted and a dispersion solution, made up of three different components, is computed on the extracted spectra. These three components are: The parameters of the fitted GIRAFFE optical model, the polynomial fit of the optical model residuals, and corrections to the slit geometry. The three components of the dispersion solution are stored in two calibration products, the dispersion solution and the setup specific slit geometry table. This splitting is done, because the creation of a new setup specific slit geometry is not always required, and may, furthermore, be a time consuming task.

An estimate of the instrument response for Argus and IFU observations may be computed using the *gistandard* recipe. However, the primary use of the created products is the instrument health monitoring and therefore the accuracy of the computed instrument response may be limited. For instance, the recipe uses the flux of all fibres as the object flux and does not select object fibres on the basis of a source detection on the reconstructed image, and it assumes no or only a simple mean sky for the background correction.

Finally, after the products of the calibration recipes are now available, the scientific observations can be processed using the last pipeline recipe *giscience*. The results are a bias corrected, and optionally, dark subtracted, extracted and wavelength calibrated spectra, corrected for fibre–to–fibre transmission and pixel–to–pixel variations. If the simultaneous calibration lamps were used during the observation, the recipe also fine tunes the wavelength scale with respect to the simultaneous calibration spectra. If object positions are available for the individual spectra the recipe also computes the barycentric, the heliocentric and the geocentric correction for each spectrum.

9 Pipeline Recipes Interfaces

This section summarises for each recipe the required raw and calibration frames, the set of offered configuration parameters, and the created products.

For each possible input frame of a recipe, the required *DO category* is given in the table of input frames. It shows also its frame type, which is one out of raw, calibration or static, and lists, whether a data frame is a required or optional input frame. Input frames which are only needed if a particular configuration parameter is used are marked as required and the name of the parameter which triggers its usage is given.

In the list of accepted configuration parameters, the fully qualified parameter name is given, as it is used, for instance, in the EsoRexconfiguration file. The common instrument prefix 'giraffe.' of the parameter names has been left out to improve the readability. The abbreviation of the parameter name, which can be used on the command line as an alias, the parameter's default value, together with a short description of its purpose is also provided.

The GIRAFFE pipeline recipes offer a large number of configuration parameters, but actually only a small subset is usually needed to process the data coming from the instrument. The parameters which are useful for normal users are therefore described in a separate section, and are explained in more detail. All other configuration parameters are intended for expert users only, and their respective default values should be kept for normal operations.

For each product that may be created by a recipe the default name of the corresponding output file is given. Note that this name has possibly been changed to a certain standard, depending on which application you are using (EsoRex or Gasgano for instance). Also the value of the FITS keyword `PRO.CATG` is listed. In the case of GIRAFFE, this value is always identical to the *DO category* assigned to this product type.

To assess the quality of a pipeline product the recipes compute quality control parameters, which are stored in the FITS headers of the recipe product files. If a recipe computes quality control parameters they are listed. The definition of the computed quality control parameters, including the algorithm used was provided by DFO (see appendix B). Detailed information on instrument quality control in general and GIRAFFE in particular is available at `http://www.eso.org/qc`.

In addition to the ordinary pipeline product file, which are all FITS files, some recipes will also create PAF files (file name extension '.paf'). These files are used in the on–line pipeline environment to ingest quality control parameters into a logging facility. Since this information is also available in the pipeline product headers these files can be safely ignored.

9.1 gimasterbias

The recipe *gimasterbias* creates a master bias frame from a set of raw bias frames. The input bias frames are combined using the selected frame stacking method. In addition the recipe produces, as optional output, a bad pixel map, which is created using a simple thresholding algorithm.

The frame stacking method is selected using the parameter `stacking-method`. The creation of the bad pixel map is controlled by the parameter `bpm-create`. For description of the recipe algorithm and how it can be controlled using the parameters listed below, see section 10.2.1.

9.1.1 Input frames

A set of raw bias frames. The number of required bias frames depends on the selected frame stacking method.

DO Category	Frame Type	Constraint	Parameter	Description
BIAS	raw	required		Raw bias frame
BAD_PIXEL_MAP	calibration	optional	mbias-bpxclean	Bad pixel map

9.1.2 Recipe parameters

Parameter	Command line alias	Values (**default**, other)	Description
stacking.method	stack-method	**average**, median, minmax, ksigma	Method used for frame combination
stacking.ksigma.low	stack-ksigmalow	**5.**	Lower threshold multiplier for stack method ksigma
stacking.ksigma.high	stack-ksigmahigh	**5.**	Upper threshold multiplier for stack method ksigma
stacking.minmax.minimum	stack-minreject	**1**	Minimum rejection level for stack method minmax
stacking.minmax.maximum	stack-maxreject	**1**	Maximum rejection level for stack method minmax
masterbias.bpm.create	bpm-create	**true**, false	Enables the creation of the bad pixel map
masterbias.bpm.factor	bpm-factor	**5.**	Readout noise multiplier defining the range of valid pixel values for the bad pixel detection

9.1.3 Detailed parameter description

stacking.method: The parameter is used to select the method for combining the input frames. For the stacking of frames, the following methods are available:

average: The mean pixel value of all input frames is computed for each pixel. The minimum number of input frames should be larger than 2.

median: The median pixel value of all input frames is computed for each pixel. The required minimum number of input frames is 3.

minmax: Is a minimum–maximum rejection. The values of a pixel in all input frames are sorted. The *stacking.minmax.minimum* lowest and the *stacking.minmax.maximum* highest pixel values are rejected. From the remaining pixel values the mean is computed. The method requires that the total number of pixel values to be rejected, is smaller than the number of input frames.

ksigma: The κ–σ clipping method. For each pixel, the median pixel value of all input frames and the median based standard deviation σ is computed. Pixel values outside a range around the median

value, which is defined by σ times *stacking.ksigma.low* and *stacking.ksigma.high*, respectively, are rejected. The remaining pixel values are averaged.

stacking.minmax.minimum: The number of low pixel values to be rejected when using the stacking method *minmax*. The number must be larger than 0.

stacking.minmax.maximum: The number of high pixel values to be rejected when using the stacking method *minmax*. The number must be larger than 0.

stacking.ksigma.low: Factor defining the lower boundary of the κ–σ rejection range.

stacking.ksigma.high: Factor defining the upper boundary of the κ–σ rejection range.

9.1.4 Extra recipe parameters

The following table summarises extra recipe parameters, which are not intended to be used for normal pipeline processing, but may be useful for expert users and pipeline maintainers.

Parameter	Command line alias	Values (**default**, other)	Description
masterbias.overscan.remove	mbias-oscremove	true, **false**	Remove prescan and overscan regions from the created master bias frame
masterbias.badpixel.clean	mbias-bpxclean	true, **false**	Correct master bias frame for bad pixels
masterbias.bpm.fraction	bpm-frac	**0.15**	Maximum fraction of pixels which may be flagged as 'bad'

9.1.5 Product frames

The following product(s) can be created by the recipe:

Default file name	PRO.CATG	Description
master_bias.fits	MASTER_BIAS	Master bias frame
bad_pixel_map.fits	BAD_PIXEL_MAP	Bad pixel map

MASTER_BIAS: The master bias contains the combined pixel values, in ADU, of the raw bias frames, with respect to the used frame stacking method. In order to be used in subsequent pipeline recipes, the prescan and overscan regions must still be present, so that the master bias frame has the same format as any raw frame.

BAD_PIXEL_MAP: The bad pixel map is an image with the same format (i.e. number of pixels along X and Y) as the master bias frame. Pixels which were flagged as "bad" by the recipe have a non–zero value.

9.1.6 Quality control parameters

The following quality control parameters are available for the master bias product:

```
QC.BIAS.MASTER.MEDIAN
QC.BIAS.MASTER.MEAN
QC.BIAS.MASTER.RMS
QC.OUT1.RON.RAW
QC.OUT1.RON.MASTER
QC.OUT1.STRUCT.X
QC.OUT1.STRUCT.Y
```

For a detailed description of their definition see appedix B.

9.2 gimasterdark

The recipe *gimasterdark* creates a master dark frame from a set of raw dark frames. The input dark frames are combined using the selected frame stacking method.

The frame stacking method is selected using the parameter `stacking-method`. For description of the recipe algorithm and how it can be controlled using the parameters listed below, see section 10.2.2.

9.2.1 Input frames

A set of raw dark frames. The number of required dark frames depends on the selected frame stacking method.

DO Category	Frame Type	Constraint	Parameter	Description
DARK	raw	required		Raw bias frame
BAD_PIXEL_MAP	calibration	optional	mbias-bpxclean	Bad pixel map

9.2.2 Recipe parameters

Parameter	Command line alias	Values (**default**, other)	Description
stacking.method	stack-method	average, **median**, minmax, ksigma	Method used for frame combination
stacking.ksigma.low	stack-ksigmalow	**5.**	Lower threshold multiplier for stack method ksigma
stacking.ksigma.high	stack-ksigmahigh	**5.**	Upper threshold multiplier for stack method ksigma
stacking.minmax.minimum	stack-minreject	**1**	Minimum rejection level for stack method minmax
stacking.minmax.maximum	stack-maxreject	**1**	Maximum rejection level for stack method minmax

Parameter	Command line alias	Values (**default**, other)	Description

9.2.3 Detailed parameter description

stacking.method: The parameter is used to select the method for combining the input frames. For the stacking of frames, the following methods are available:

average: The mean pixel value of all input frames is computed for each pixel. The minimum number of input frames should be larger than 2.

median: The median pixel value of all input frames is computed for each pixel. The required minimum number of input frames is 3.

minmax: Is a minimum–maximum rejection. The values of a pixel in all input frames are sorted. The *stacking.minmax.minimum* lowest and the *stacking.minmax.maximum* highest pixel values are rejected. From the remaining pixel values the mean is computed. The method requires that the total number of pixel values to be rejected, is smaller than the number of input frames.

ksigma: The κ–σ clipping method. For each pixel, the median pixel value of all input frames and the median based standard deviation σ is computed. Pixel values outside a range around the median value, which is defined by σ times *stacking.ksigma.low* and *stacking.ksigma.high*, respectively, are rejected. The remaining pixel values are averaged.

stacking.minmax.minimum: The number of low pixel values to be rejected when using the stacking method *minmax*. The number must be larger than 0.

stacking.minmax.maximum: The number of high pixel values to be rejected when using the stacking method *minmax*. The number must be larger than 0.

stacking.ksigma.low: Factor defining the lower boundary of the κ–σ rejection range.

stacking.ksigma.high: Factor defining the upper boundary of the κ–σ rejection range.

9.2.4 Extra recipe parameters

The following table summarises extra recipe parameters, which are not intended to be used for normal pipeline processing, but may be useful for expert users and pipeline maintainers.

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.remove	remove-bias	**true**, false	Turn bias removal on and off
biasremoval.method	bsremove-method	UNIFORM, PLANE, **PROFILE**, CURVE, MASTER, ZMASTER, PROFILE+CURVE, MASTER+PLANE, MASTER+CURVE, ZMASTER+PLANE, ZMASTER+CURVE	Method to use for correcting the bias.
biasremoval.areas	bsremove-areas	**5:40:0:4095**	Specification of image areas to be used for the bias correction
biasremoval.sigma	bsremove-sigma	**2.5**	Sigma clipping threshold factor
biasremoval.iterations	bsremove-niter	**5**	Sigma clipping number of iterations
biasremoval.fraction	bsremove-mfrac	**0.8**	Sigma clipping minimum fraction of accepted points
biasremoval.xorder	bsremove-xorder	**1**	Order of polynomial fit along X (method CURVE only)
biasremoval.yorder	bsremove-yorder	**1**	Order of polynomial fit along Y (method CURVE only)
biasremoval.xstep	bsremove-xstep	**1**	Sampling step along X (method CURVE only)
biasremoval.ystep	bsremove-ystep	**1**	Sampling step along Y (method CURVE only)

9.2.5 Product frames

The following product(s) can be created by the recipe:

Default file name	PRO.CATG	Description
master_dark.fits	MASTER_DARK	Master dark frame

MASTER_DARK: The master dark contains the combined pixel values of the bias corrected raw dark frames, normalized to 1 second. The pixel values therefore are in units of ADU/s. How the pixel values have been combined depends on the selected stacking method. If a bad pixel mask was included in the input set of frames, the pixel values corresponding to the locations of bad pixels are set to 0.

9.2.6 Quality control parameters

The following quality control parameters are available for the master dark product:

QC.DARK.CURRENT
QC.GLOOM.LEVEL
QC.GLOOM.XPOS
QC.GLOOM.YPOS

Note that the QC parameters describing the prominent glow feature in the upper right part of the old CCD *Bruce* have no relevance for observations carried out with the new CCD *Carreras*, which has been installed in May 2008.

For a detailed description of their definition see appedix B.

9.3 gimasterflat

The purpose of the recipe *gimasterflat* is the detection and the tracing of the spectra on a fibre flat–field image for a given fibre setup, i.e. creating the fibre localisation. The fibre setup may be specified on the command line using the command line options `fiber-splist` or `fiber-nspec` and by adding an already existing fiber localisation centroid frame to the list of input frames.

In the latter case the passed localisation centroid frame is used as a reference for the fibre setup, which is essentially copied from it. But the positions of the fibres and their widths are adjusted.

In addition the fiber setup can be taken from a setup specific slit geometry frame, if it is part of the input data set and the parameter `fiber-splist` is set to "setup". Specifying the fiber setup in this way is equivalent to manually passing the list of available fibers directly to the parameter `fiber-splist`. However the advantage is that one does not have to lookup the list of active fibers of the slit system which was used for the observation as long as a valid slit geometry frame is available. However, this is usually the case, since the setup specific slit geometry frames are included in the data packages available from the ESO archive.

If no suitable reference localisation centroid frame or setup specific slit geometry is available (e.g. an additional fibre may be broken), the command line options must be used, to explicitly specify which fibres are present in the raw flat–field image, in order to create a valid fibre localisation. The arguments passed to the command line options `fiber-splist` or `fiber-nspec` define the fibre setup, i.e. they establish the relation between a fibre signal detected on the raw flat–field image and the position of this fibre within the slit (see section 6). Using a wrong set of fibres at this stage would cause a wrong assignement of the spectrum lookup index (the column `INDEX`) in the fibre setup table (see section 6) of the product images, and may therefore lead to calibrating a science spectra with the wrong calibration spectrum.

Information on the status of the GIRAFFE fibres can be found in the section on the raw data spectral format at `http://www.eso.org/observing/dfo/quality/GIRAFFE/pipeline/pipe_gen.html`.

Furthermore, the recipe determines the parameters of the fibre profile, by fitting an analytical model of this profile to the flat field data. Apart from the fitted profile parameters, this step also provides a fiber localisation centroid and half width frame as products. These localisation products are computed from the fitted fibre profiles, but are otherwise equivalent to the standard localisation products.

In addition to the fibre localisation products, the recipe creates an extracted flat–field frame. The fibre–to–fibre transmission is computed optionally, and, if it was requested the transmission and its error are stored in two extra columns, which are added to the fibre setup table of the extracted flat–field frame. They are labeled TRANSMISSION and DTRANSMISSION respectively.

In an optional step the recipe computes a scattered light model from the inter-spectrum regions. With this method a reasonable results can only be obtained for the MEDUSA instrument mode, since in the case of IFU or Argus the inter-spectrum regions are limited to the few gaps between the individual fibre subslits. Also, for data taken before 15th of May 2008 using the CCD *Bruce*, the glow feature must be removed prior to the scattered light model computation. Usually the use of the scattered light model computed in this way is not necessary. Instead the use of the optimal extraction method should be considered.

Since the localisation results of this recipe are used in all subsequent recipes of the reduction cascade as the position and width of the spectra on the detector, it is strongly recommended to verify these results visually, for instance by overlaying the computed fibre centroid positions to the master flat field frame.

For description of the recipe algorithm and how it can be controlled using the parameters listed below, see section 10.2.3.

9.3.1 Input frames

The recipe requires one or more raw fibre flat–field frames in input, i.e. flat–field lamp exposures using one of the available fibre systems.

DO Category	Frame Type	Constraint	Parameter	Description
FIBER_FLAT	raw	required		Raw fibre flat-field lamp exposure
MASTER_BIAS	calibration	required		Master bias
MASTER_DARK	calibration	optional		Master dark
BAD_PIXEL_MAP	calibration	optional		Bad pixel map
FF_LOCCENTROID	calibration	optional	fiber-splist, fiber-nspec	Localisation centroid positions; used as template for the fibre setup
SLIT_GEOMETRY_MASTER, SLIT_GEOMETRY_SETUP	static	required		Slit geometry template
GRATING_DATA	static	required		Grating physical parameters

9.3.2 Recipe parameters

Parameter	Command line alias	Values (**default**, other)	Description
fibers.spectra	fiber-splist		Comma separated list of fiber positions in the slit (ranges may be used), or “setup”
fibers.nspectra	fiber-nspec	**136**	Number of spectra to be detected.

Parameter	Command line alias	Values (**default**, other)	Description
localization.ewidth	sloc-ewidth	**1.**	Localization detection extra width. Extra space added at both sides of the detected spectra.
localization.ywidth	sloc-ywidth	**-1**	Full width [pxl] of the equilizing filter (distance between two adjacent fibers)
localization.noise	sloc-noise	**7.**	Detection threshold multiplier
localization.yorder	sloc-yorder	**4**	Order of Chebyshev polynomial fit, used to model the fibre centroid positions
localization.worder	sloc-worder	**2**	Order of Chebyshev 2D polynomial fit of the fibre width
psf.model	psf-model	psfexp, **psfexp2**, gaussian	Analytical model of the fibre profile
psf.parameters.fit	psf-prmfit	true, **false**	Use a 2d Chebyshev polynomial model to fit the profile parameters, instead of the standard 1d model.
psf.parameters.yorder	psf-yorder	**4**	Order of the Chebyshev polynomial model fitted to the profile parameters along the dispersion direction.
psf.parameters.worder	psf-yorder	**4**	Order of the Chebyshev polynomial model fitted to the profile parameters along the spatial direction, if a 2d model is used.
psf.parameters.sigma	psf-sigma	**3.5**	Sigma threshold for removing data points when fitting the fibre profile parameters.
psf.parameters.iterations	psf-niter	**10**	Number of iterations used for fitting the fibre profile parameters.
psf.parameters.fraction	psf-mfrac	**0.8**	Minimum fraction of data points kept when fitting the fibre profile parameters.

Parameter	Command line alias	Values (**default**, other)	Description
extraction.method	extr-method	**SUM**, HORNE, OPTIMAL	Spectrum extraction method
extraction.ron	extr-ron	**-1.**	New bias sigma (RON) value for bias and dark corrected image
extraction.psf.model	extr-psfmodel	psfexp, **psfexp2**	Analytical model of the fibre profile
extraction.psf.sigma	extr-psfsigma	**7.**	Sigma clipping threshold used for rejecting data points while fitting the fibre profile to the data.
extraction.psf.iterations	extr-psfniter	**2**	Maximum number of iterations used for fitting the fibre profile.
extraction.horne.extrawidth	extr-hewidth	**2**	Number of extra pixels added to the fibre half-width
extraction.horne.mingood	extr-hmingood	**3**	Minimum number of data points kept for fitting the fibre profile
extraction.optimal.fraction	extr-omfrac	**0.9**	Minimum fraction of data points kept for fitting the fibre profiles of a wavelength bin
extraction.optimal.wfactor	extr-owfactor	**3.**	Scale factor of the fibre half-width
extraction.optimal.bkgorder	extr-obkgorder	**2**	Order of the polynomial background model.
masterflat.transmission	transmission	**true**, false	Enables the relative fiber transmission computation
masterflat.slight	slight	true, **false**	Enables the scattered light model computation

9.3.3 Detailed parameter description

fiber.spectra: The argument of this parameter is a comma separated list of fibre positions in the slit (cf. section 6.1). The fibre positions given as argument are assigned to the spectra detected by the recipe, starting from the leftmost pixel column of a flat–field frame, thereby defining the fibre setup. The argument may also contain fiber position ranges, i.e. two numbers separated by a hyphen, like in the following example:

```
-fiber-splist=1-191,193-302,304-318
```

If the list should be generated from the slit geometry of this instrument setup, the string `setup` must be given as parameter value, and an appropriate slit geometry frame (SLIT_GEOMETRY_SETUP) must be present in the input data set.

fiber.nspectra: This parameter can be used as an alternative to the parameter `fiber.spectra`, in the case that no fibres are missing in the range from 1 to the number of spectra *N* provided as the parameter's argument. The recipe simply assigns the fibre positions 1 to *N* to the first *N* detected spectra.

localization.ewidth: The provided number is the extra space in units of pixels which is added twice (at the left and right spectrum border) to the detected spectrum width.

localization.ywidth: The given number defines the width of the fibre detection window. The given value should be of the order of the distance between two adjacent fibres, i.e. slightly larger than the fibre width.

localization.noise: The parameter controls the sensitivity of the spectrum detection task. It is used as a multiplier for the noise of the pixel value. The pixel is accepted as a spectrum candidate pixel if the background subtracted pixel value is larger than the noise multiplied by the parameter's argument.

psf.model: Name of the analytical fibre profile model. Available are a gaussian profile and the two exponential profiles

psfexp: $P(x) = Ae^{-(|x-x_0|^{\varepsilon}/\omega)} + B$

psfexp2: $P(x) = Ae^{-(|x-x_0|/\omega)^{\varepsilon}} + B$

The model is fitted to the observed data to determine the profile parameters.

psf.parameters.fit: Selects the type of polynomial model used to fit the fibre profile parameters (A, x_0, ε, ω, B). If it is set a (global) 2-dimensional Chebyshev polynomial model is used to model the profile parameters instead of the default 1-dimensional Chebyshev polynomial model, which fits the profile parameters for each fibre individually.

psf.parameters.yorder: Order of the polynomial model along the dispersion axis which is fitted to the profile parameters determined for each wavelength bin. It is used for both, the 1-dimensional and the 2-dimensional case.

psf.parameters.worder: Order of the polynomial model along the spatial axis which is fitted to the profile parameters determined for each wavelength bin. It is only used in the 2-dimensional case.

psf.parameters.sigma: The factor κ of the κ-σ clipping algorithm used to reject data points when fitting the profile parameters.

psf.parameters.iterations: The maximum number of iterations of the κ-σ clipping algorithm.

psf.parameters.fraction: The minimum fraction f of data points which are kept for fitting the profile parameters, i.e. the κ-σ clipping algorithm will reject at most $1-f$ data points.

extraction.method: Name of the method to be used for extracting the spectra. Available are a standard extraction, a simple sum along the slit, an implementation of Horne's method for a known profile function [RD6], and an optimal extraction algorithm, which extracts the flux for all fibres and the background simultaneously. For the latter two methods the extracted fluxes are already background subtracted. In the case of the optimal extraction the input data should not be corrected for the scattered light. In addition this method naturally treats a possible cross-talk between adjacent fibres correctly.

extraction.ron: Usually the read-out noise is taken from the FITS header of the bias subtracted input frame. This parameter may be used to override the value in the FITS header.

extraction.psf.model: The name of the fibre profile analytical model to use for the spectrum extraction. Note that this parameter should match the model which is used to obtain the fibre profile parameters.

extraction.psf.sigma: This is Horne's sigma (cf. RD6). The σ threshold is used to reject bad pixels and cosmics during the extraction of the spectra. Data points are rejected if the residual of a data point is larger than σ times the data point's expected variance (i.e. computed from the profile model). Rejected data points are ignored and not replaced by the model.

extraction.psf.iterations: The maximum number of iterations of cleaning cycle of the Horne and optimal extraction method.

extraction.horne.extrawidth: Number of extra pixels by which the extraction slit is enlarged in case of using the Horne extraction. The pixels are added to the half-width of the fibres.

extraction.horne.mingood: The minimum number of pixels that have to be kept for the spectrum extraction. This sets the lower limit of pixels which can be rejected by the cleaning cycle.

extraction.optimal.fraction: The minimum number of pixels that have to be kept for the spectrum extraction. This sets the lower limit of pixels which can be rejected by the cleaning cycle. If the optimal extraction is used, this lower limit has to be specified as a fraction, since all fibres are treated simultaneously and not each fibre separately.

extraction.optimal.wfactor: This factor is used to set the maximum distance between the current spectrum and the neighbour furthest away which is assumed to contribute to the observed flux by cross-talk. All spectra with smaller distances than this factor times the current spectrum's half-width are taken into account during the flux computation.

extraction.optimal.bkgorder: The optimal extraction determines the background (scattered light) simultaneously with the computation of the fluxes of all spectra. For this it assumes a background that can be modeled by a Chebyshev polynomial. The parameter sets the order of this polynomial model.

9.3.4 Extra recipe parameters

The following table summarises extra recipe parameters, which are not intended to be used for normal pipeline processing, but may be useful for expert users and pipeline maintainers.

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.remove	remove-bias	**true**, false	Turn bias removal on and off
biasremoval.method	bsremove-method	UNIFORM, PLANE, **PROFILE**, CURVE, MASTER, ZMASTER, PROFILE+CURVE, MASTER+PLANE, MASTER+CURVE, ZMASTER+PLANE, ZMASTER+CURVE	Method to use for correcting the bias.

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.areas	bsremove-areas	**5:40:0:4095**	Specification of image areas to be used for the bias correction
biasremoval.sigma	bsremove-sigma	**2.5**	Sigma clipping threshold factor
biasremoval.iterations	bsremove-niter	**5**	Sigma clipping number of iterations
biasremoval.fraction	bsremove-mfrac	**0.8**	Sigma clipping minimum fraction of accepted points
biasremoval.xorder	bsremove-xorder	**1**	Order of polynomial fit along X (method CURVE only)
biasremoval.yorder	bsremove-yorder	**1**	Order of polynomial fit along Y (method CURVE only)
biasremoval.xstep	bsremove-xstep	**1**	Sampling step along X (method CURVE only)
biasremoval.ystep	bsremove-ystep	**1**	Sampling step along Y (method CURVE only)
localization.mode	sloc-mode	**all**, siwc	Use all or only the 5 SIWC spectra
localization.start	sloc-start	**-1**	Pixel row from where the fiber detection starts. The default refers to the image center.
localization.retries	sloc-retries	**10**	Initial localization detection xbin retries.
localization.binsize	sloc-binsize	**-1**	Initial localization detection xbin size
localization.center	sloc-center	**centroid**, hwidth	Method used for mask center computation
localization.normalize	sloc-norm	true, **false**	Enable spectrum normalization along the dispersion axis
localization.threshold	sloc-threshold	**local**, global, row	Selects thresholding algorithm: local, global, row (*experimental: requires* localization.threshold *less than 0*).
localization.ron	sloc-ron	**-1.**	New bias sigma (RON) value for dark subtraction
localization.sigma	sloc-sigma	**2.5**	Localization clipping: sigma threshold
localization.iterations	sloc-niter	**5**	Localization clipping: number of iterations

Parameter	Command line alias	Values (**default**, other)	Description
localization.fraction	sloc-mfrac	**0.9**	Localization clipping: minimum fraction
psf.normalize	psf-norm	true,**false**	Use normalized pixel values
psf.binsize	psf-binsize	**64**	Size of the wavelength bin used to improve the signal-to-noise ratio for fitting individual fibre profiles
psf.maxwidth	psf-maxwidth	**16.**	Maximum allowed width of the fibre profile
psf.width	psf-width	**0.**	Initial fibre profile width parameter
psf.exponent	psf-exponent	**-3.**	Initial fibre profile exponent parameter. It is not fitted if it is larger than 0.
psf.profile.iterations	psf-pfniter	**120**	Maximum number of iterations used for the fit of the fibre profile
psf.profile.ntest	psf-pfntest	**7**	Maximum number of tests used for the fibre profile fit
psf.profile.dchisquare	psf-pfdchisq	**0.001**	Minimum χ^2 difference used for fitting the fibre profile
slight.model.name	slight-model	**polynom**, polyfrac	Name of the scattered light model to use.
slight.model.order	slight-order	**4,2**	Order of the 2d polynomial model along the dispersion and the spatial axis.
slight.xstep	slight-xstep	**10**	Inter-spectrum region sampling step along the dispersion direction
slight.ystep	slight-ystep	**1**	Inter-spectrum region sampling step along the spatial direction
slight.xslice	slight-xslice	**none**	Inter-spectrum region sampling step along the dispersion direction for a specific region. This overrides 'xstep' for the given region sampling step along the spatial direction
slight.ewidth	slight-ewidth	**0.5**	Extra width in units of pixels added to both sides of a spectrum trace
slight.iswidth	slight-iswidth	**2**	Minimum width in units of pixels required for inter-spectrum regions

Parameter	Command line alias	Values (**default**, other)	Description
slight.istrim	slight-istrim	**true**, false	Disables the use of the first and last inter-spectrum region
slight.phffcorrection	slight-phff	true, **false**	Use photometric flat field correction
slight.remove	slight-remove	true, **false**	Remove scattered light from the input frame

9.3.5 Product frames

Default file name	PRO.CATG	Description
master_fiber_flat.fits	MASTER_FIBER_FLAT	Average of all raw flat–field frames
ff_loccentroid.fits	FF_LOCCENTROID	Fibre localisation; spectrum centroid position on the CCD
ff_locwidth.fits	FF_LOCWIDTH	Fibre localisation; spectrum half-width on the CCD
ff_locfit.fits	FF_LOCFIT	Fibre localisation; fit coefficients of standard localisation (unused)
ff_psfcentroid.fits	FF_PSFCENTROID	Fibre localisation; Fitted fibre profile center
ff_psfwidth.fits	FF_PSFWIDTH	Fibre localisation; Fitted fibre profile half-width
ff_psffit.fits	FF_PSFFIT	Fibre localisation; fit coefficients of fibre profile localisation (unused)
fiber_profile.fits	FIBER_PROFILE	Fitted fibre profile parameters
ff_extspectra.fits	FF_EXTSPECTRA	Extracted and normalised flat–field lamp spectra
ff_exterrors.fits	FF_EXTERRORS	Statistical error of the extracted flat–field lamp spectra
ff_extpixels.fits	FF_EXTPIXELS	Number of pixels contributing a each wavelength bin of the extracted flat–field lamp spectra (Only for extraction method SUM and HORNE.)
ff_exttraces.fits	FF_EXTTRACES	Centroid position in pixels of the extracted fluxes for each wavelength bin
ff_extmodel.fits	FF_EXTMODEL	Profile model used for the spectrum extraction (Only for extraction method OPTIMAL.)

The following products are stored using the extracted frame format (cf. section 6.1): FF_LOCCENTROID, FF_LOCWIDTH, FF_PSFCENTROID, FF_PSFWIDTH, FF_EXTSPECTRA, FF_EXTERRORS, FF_EXTPIXELS and FF_EXTTRACES.

MASTER_FIBER_FLAT: The master flat–field frame is the average of the bias subtracted raw flat–field frames. The master flat–field frame has the same format as a raw flat–field frame, but the prescan and overscan regions have been removed.

FF_LOCCENTROID: The image is used to store the fitted fibre centroid positions on the detector for each wavelength bin (standard localisation).

FF_LOCWIDTH: The fitted fibre width for each wavelength bin stored in an image (standard localisation).

FF_LOCFIT: The fitted coefficients of the Chebyshev polynomials used to model the fibre centroid positions (standard localisation).

FF_PSFCENTROID: The image is used to store the fitted fibre centroid positions on the detector for each wavelength bin. The fibre centroids were determined from the fit of the fibre profile.

FF_PSFWIDTH: The fitted fibre width for each wavelength bin stored in an image. The fibre half-widths were determined from the fit of the fibre profile.

FF_PSFFIT: The fitted coefficients of the Chebyshev polynomials used to model the fibre centroid positions. The coefficients refer to the centroids determined from the fibre profile fit.

FIBER_PROFILE: The fitted fibre profile parameters for each fibre and wavelength bin. This information is needed for the spectrum extraction methods based on a fibre profile fit.

FF_EXTSPECTRA: The extracted flat–field lamp fluxes normalised to the mean flux of all extracted flat–field lamp spectra. The signal–to–noise ratio of the extracted spectra may simply be computed, dividing the extracted spectra by the error spectra of the FF_EXTERRORS product.

FF_EXTERRORS: The statistical error (standard deviation) of the extracted and normalised fluxes for each wavelength bin.

FF_EXTPIXELS: The number of pixels which contributed to the extracted fluxes of each wavelength bin.

FF_EXTTRACES: The centroid position of the extracted fluxes for each wavelength bin.

FF_EXTMODEL: The fibre profile model used to extract the observed fluxes. This frame is only available if the optimal extraction has been used. It is basically a simulation of the input frame and can be compared with that to assess the quality of the spectrum extraction.

9.3.6 Quality control parameters

Quality control parameters are available for the master flat–field, the localisation centroid and width frames, and the extracted spectra frame.

```
QC.FIBRE.CENTROID.DIFF
QC.FIBRE.CENTROID.RMS
QC.FIBRE.WIDTH.MEAN
QC.FIBRE.WIDTH.RMS
QC.FIBRE.TRANS.MEDIAN
QC.FIBRE.TRANS.RMS
QC.FIBRE.TRANS.ERROR
QC.OUT1.MEAN.RAW
QC.OUT1.NSAT.RAW
QC.LAMP.EFFIC
QC.LAMP.EFFIC1
```

For a detailed description of their definition see appedix B.

9.4 giwavecalibration

The wavelength calibration recipe *giwavecalibration* computes the dispersion solution, and, optionally, a slit geometry table for the fibre setup in use.

A complete GIRAFFE dispersion solution consists of an optical model, a polynomial fit of the optical model residuals, and the correction of individual fibre offsets. The first two components are stored in the dispersion solution product frame, whereas the latter is a correction term to the slit geometry giving the slit geometry table specific for the current instrument and fibre setup.

The recipe uses an existing dispersion solution to predict the positions of the ThAr lines on the detector. This initial guess is either taken from an existing, complete wavelength solution (optical model, residuals and slit geometry), or from the grating data table. Using the average optical model from the grating table usually requires an adjustment of the recipe parameters (e.g. increasing the search window width; using 20 instead of the default 10 should be sufficient), and more iterations (the number of iterations is given by the number of search windows specified) are necessary to get to a good dispersion solution.

If a new slit geometry table should be created, the option `wcal-slit` (see below) must be enabled. Whether the slit geometry should be updated can be easily seen from the rebinned arc lamp frame. A "bad" slit geometry would cause small offsets (jumps) for individual fibres, or small groups of fibres in the otherwise straight calibration line. Re-creating a slit geometry can be time consuming, because it involves a repeated rebinning of the extracted arc lamp spectra. The slit geometry should be iterated until a negligible "radial velocity" (see column RV in the created slit geometry table) for all fibres.

How to start, if there is no dispersion solution nor a setup specific slit geometry available? The strategy in this case is to start from the average optical model in the grating table and the slit geometry template. In this very first run a search window width of 20 should still be sufficient. The number of iterations for that search window width should not be too small, and also the number of iterations for the slit geometry computation should not be choosen too small (about 8). In rare cases one might have to manually adjust the initial slit offset. This will create the first full dispersion solution for the instrument and fibre setup used. It can then be refined in subsequent iterations, i.e. re-running the recipe *giwavecalibration* using the new dispersion solution and slit geometry as initial guess.

For description of the recipe algorithm and how it can be controlled using the parameters listed below, see section 10.2.4.

9.4.1 Input frames

The recipe expects a single arc lamp spectrum as input raw frame. In addition, it needs a dispersion solution, which is used as initial guess. Either an already existing, full dispersion solution (obtained from a previous run, or DFO packages for instance), or the average optical model stored in the grating data table can be used. The latter is used whenever no dispersion solution is given as input.

The slit geometry template is only needed in the first run after the fibre setup has been changed such that it contains fibres which are not covered by the instrument setup specific slit geometry table. For instance if the number of fibres increased, or a dead fiber was resurrected. In all other cases an appropriate setup specific slit geometry table should be used.

DO Category	Frame Type	Constraint	Parameter	Description
ARC_SPECTRUM	raw	required		ThAr arc lamp spectrum
MASTER_BIAS	calibration	optional	remove-bias	Master bias
MASTER_DARK	calibration	optional		Master dark
FF_LOCCENTROID, FF_PSFCENTROID	calibration	required		Fibre localisation, centroid position
FF_LOCWIDTH, FF_PSFWIDTH	calibration	required		Fibre localisation, spectrum width
DISPERSION_SOLUTION	calibration	optional		Dispersion solution used as initial guess
SLIT_GEOMETRY_SETUP	calibration	required		Setup specific slit geometry table. This is required for normal operations, but may be replace by a slit geometry template for a bootstrap run.
SLIT_GEOMETRY_MASTER	static	optional	wcal–slit	Slit geometry template, this has to be used if a slit geometry for a new fibre setup must be created. In this case no setup specific slit geometry table must be provided
LINE_CATALOG	static	required		ThAr line catalog
GRATING_DATA	static	required		Grating physical parameters, and mean optical model parameters
LINE_MASK	static	optional	wcal–slit	Line mask for this setup. This is only needed if an new slit geometry should be created.

9.4.2 Recipe parameters

Parameter	Command line alias	Values (**default**, other)	Description
wlcalibration.line.widths	wcal-lswidth	**10,10,10,10,10**	List of window widths [pxl] used for line detection and fit
wlcalibration.line.wrange	wcal-lswrange	**0.,0.**	Selects only lines within the given wavelength range [nm]
wlcalibration.line.residuals	wcal-lfres	**auto**, enable, disable	Take optical model residuals into account during line detection
wlcalibration.slit.offset	wcal-soffset	**setup**	Initial slit position offsets along the x and y direction and rotation angle
wlcalibration.opt.solution	wcal-omsol	**true**, false	Enables optical model parameter fitting
wlcalibration.opt.flags	wcal-omflags	fcoll, gcam, theta, **sdx, sdy, sphi**,	List of flags defining the set of free parameters used for fitting the optical model
wlcalibration.psf.sigma	wcal-xwsigma	**1.25**	PSF width fit sigma clipping factor
rebinning.xresiduals	rbin-xresid	**true**, false	Take optical model residuals into account during spectrum rebinning
rebinning.range	rbin-range	**setup**, common	Rebinning range
sgcalibration.iterations	scal-cniter	**1**	Maximum number of iterations for the cross-correlation to determine residual wavelength shifts
sgcalibration.zmax	scal-zmax	**10000.**	Maximum allowed pixel value. To be effective it must be larger than 0.
wcal.rebin	wcal-rebin	**true**, false	Rebin extracted arc-lamp spectra
wcal.slit	wcal-slit	true, **false**	Enables the slit geometry calibration.

9.4.3 Detailed parameter description

wlcalibration.line.widths: The parameter's argument is a comma separated list of line search window widths in units of pixels. For each value in the list, the wavelength calibration is performed using the computed dispersion solution from the previous iteration as initial guess. The calibration lines selected from the line catalog for each iteration takes the search window width to use into account, i.e. smaller search windows will use more lines for the calibration. The search window size should not be choosen smaller than 10 pixels.

wlcalibration.line.wrange: The parameters sets the wavelength range (minimum, maximum) wavelength for the selection of calibration lines from the line catalog. The default value "0.,0." tells the recipe to use the wavelength limits stored in the grating data table for this instrument setup.

wlcalibration.line.residuals: The parameter controls whether the fitted optical model residuals are used for predicting the positions of the calibration lines on the detector. If the parameter is set to "auto", the residuals are taken into account, if they are present in the input dispersion solution passed as an initial

guess to the recipe. This is usually the case if a full dispersion solution obtained from a previous run is used as initial guess. Setting the parameter value to "disable" the residuals are not used, and setting it to "enable" will always use the residuals (this, of course, requires that the fitted residuals are present in the initial guess dispersion solution).

wlcalibration.slit.offset: The parameter allows to override the initial values of the slit offsets in X and Y and the slit rotation angle. The slit rotation angle is in units radians.

wlcalibration.opt.flags: Selects the optical model parameters which will be used as free parameters during the optical model fit. The parameter argument may be a comma separated list of the values: fcoll, gcam, theta, sdx, sdy, and sphi, which correspond to the collimator focal length, camera magnification, grating angle, slit offset along a and y, and the slit rotation, respectively. See also section 10.1.5.

wlcalibration.psf.sigma: The parameter can be used to control the line rejection based on the line's FWHM. The parameters argument is the multiplied by the uncertainty of the line's FWHM to define a rejection threshold. A line is rejected, if the difference between the line's FWHM and the 2–dimensional fit of the line FWHM across the detector is larger the the rejection threshold.

rebinning.range Selects the wavelength range used for the spectrum rebinning. Setting the parameter to "setup" will use the wavelength range as it is defined in the grating data table for the instrument setup, whereas setting it to common will use the wavelength interval which is common to all allocated fibres.

sgcalibration.iterations: The parameter defines how often the cross–correlation of the rebinned spectra and the line mask is repeated, using the slit geometry from the previous iteration as input. The slit geometry calibration should be repeated until the computed residual fibre offset, expressed in terms of radial velocity, is negligible for all fibers.

sgcalibration.zmax: This is a truncation parameter for the line intensities in order to avoid that the cross–correlation is affected by heavily different line intensities, i.e. the result is biased because of a dominating, bright line. To find the proper truncation value a few tries may be necessary. The truncation value is in units of ADU and is applied to the rebinned spectra.

9.4.4 Extra recipe parameters

The following table summarises extra recipe parameters, which are not intended to be used for normal pipeline processing, but may be useful for expert users and pipeline maintainers.

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.remove	remove-bias	**true**, false	Turn bias removal on and off
biasremoval.method	bsremove-method	UNIFORM, PLANE, **PROFILE**, CURVE, MASTER, ZMASTER, PROFILE+CURVE, MASTER+PLANE, MASTER+CURVE, ZMASTER+PLANE, ZMASTER+CURVE	Method to use for correcting the bias.
biasremoval.areas	bsremove-areas	**5:40:0:4095**	Specification of image areas to be used for the bias correction
biasremoval.sigma	bsremove-sigma	**2.5**	Sigma clipping threshold factor
biasremoval.iterations	bsremove-niter	**5**	Sigma clipping number of iterations
biasremoval.fraction	bsremove-mfrac	**0.8**	Sigma clipping minimum fraction of accepted points
biasremoval.xorder	bsremove-xorder	**1**	Order of polynomial fit along X (method CURVE only)
biasremoval.yorder	bsremove-yorder	**1**	Order of polynomial fit along Y (method CURVE only)
biasremoval.xstep	bsremove-xstep	**1**	Sampling step along X (method CURVE only)
biasremoval.ystep	bsremove-ystep	**1**	Sampling step along Y (method CURVE only)
extraction.method	extr-method	**SUM**, HORNE, OPTIMAL	Spectrum extraction method
extraction.ron	extr-ron	**-1.**	New bias sigma (RON) value for bias and dark corrected image
extraction.psf.model	extr-psfmodel	psfexp, **psfexp2**	Analytical model of the fibre profile
extraction.psf.sigma	extr-psfsigma	**7.**	Sigma clipping threshold used for rejecting data points while fitting the fibre profile to the data.
extraction.psf.iterations	extr-psfniter	**2**	Maximum number of iterations used for fitting the fibre profile.
extraction.horne.extrawidth	extr-hewidth	**2**	Number of extra pixels added to the fibre half-width

Parameter	Command line alias	Values (**default**, other)	Description
extraction.horne.mingood	extr-hmingood	**3**	Minimum number of data points kept for fitting the fibre profile
extraction.optimal.fraction	extr-omfrac	**0.9**	Minimum fraction of data points kept for fitting the fibre profiles of a wavelength bin
extraction.optimal.wfactor	extr-owfactor	**3.**	Scale factor of the fibre half-width
extraction.optimal.bkgorder	extr-obkgorder	**2**	Order of the polynomial background model.
wlcalibration.line.separation	wcal-lssep	**0.9**	Factor used to compute the minimum line separation from the window width
wlcalibration.line.fluxratio	wcal-lsfxratio	**50.**	Selects only lines whose neighbours have a relative intensity less than 1. / fluxratio
wlcalibration.line.brightness	wcal-lsbright	**0.**	Selects lines having an intensity greater or equal to the given intensity
wlcalibration.line.count	wcal-lscount	**-80**	Sets the minimum number of lines to select; selected are lines with the highest nominal intensity. A value of 0 turns this selection off. If the value is less than 0 the selection is skipped if the line list does not contain enough lines
wlcalibration.line.model	wcal-lfmodel	**psfexp**, psfexp2	Line profile model
wlcalibration.line.threshold	wcal-lfthreshold	**1.**	Line detection threshold during the line fitting (multiple of bias sigma)
wlcalibration.line.offset	wcal-lfoffset	**10.**	Maximum allowed difference between the fitted and raw line peak position
wlcalibration.line.iterations	wcal-lfniter	**50**	Line detection fit maximum number of iterations
wlcalibration.line.tests	wcal-lfntest	**7**	Line detection fit maximum number of tests
wlcalibration.line.dchisquare	wcal-lfdchisq	**0.0001**	Line detection fit minimum chi-square difference
wlcalibration.line.rwidthratio	wcal-lfreswid	**0.5**	Line width/resolution width factor
wlcalibration.line.exponent	wcal-lfexpwid	**-3.**	Exponential line profile exponent; it will not be fitted if it is larger than 0.

Parameter	Command line alias	Values (**default**, other)	Description
wlcalibration.opt.model	wcal-ommodel	xoptmod, **xoptmod2**	Optical model
wlcalibration.opt.direction	wcal-omdir	-1, **1**	Dispersion direction flag
wlcalibration.opt.subslits	wcal-omsslits	true, **false**	Controls subslit geometry usage in the optical model fit; subslits are used if set to 'true'.
wlcalibration.opt.iterations	wcal-omniter	**50**	Optical model fit maximum number of iterations
wlcalibration.opt.tests	wcal-omntest	**7**	Optical model fit maximum number of tests
wlcalibration.opt.dchisquare	wcal-omdchisq	**0.0001**	Optical model fit minimum chi-square difference
wlcalibration.psf.iterations	wcal-xwniter	**10**	PSF width fit sigma clipping maximum number of iterations
wlcalibration.psf.fraction	wcal-xwmfrac	**0.9**	PSF width fit sigma clipping minimum fraction of points accepted/total
wlcalibration.psf.order	wcal-xworder	**2,2**	X and Y polynomial orders for PSF x-width Chebyshev fit
wlcalibration.wsol.sigma	wcal-wssigma	**150.**	Chebyshev correction sigma clipping factor.
wlcalibration.wsol.iterations	wcal-wsniter	**10**	Chebyshev correction sigma clipping maximum number of iterations
wlcalibration.wsol.fraction	wcal-wsmfrac	**0.9**	Chebyshev correction sigma clipping minimum fraction of points accepted/total
wlcalibration.wsol.order	wcal-wsorder	**6,4**	X and Y polynomial orders for the wavelength solution Chebyshev correction
rebinning.method	rbin-method	**linear**, spline	Rebinning method
rebinning.scalemethod	rbin-scmethod	**linear**, log	Rebinning scaling method
rebinning.lstep	rbin-lstep	**0.005**	Lambda step size, only used if rebinning.scalemethod is "linear"
rebinning.size	rbin-size	**0**	Size of output rebinned spectra in terms of pixels. If it is set to 0 the size is computed from the wavelength range and lambda stepsize. If it is larger than 0 the lambda stepsize is adjusted so that the wavelength range is covered by the given number of pixels.

Parameter	Command line alias	Values (**default**, other)	Description
sgcalibration.cc.step	scal-cstep	**-0.005**	Cross-correlation step
sgcalibration.cc.domain	scal-cdomain	**0.,0.**	Restricts the cross-correlation to the given domain
sgcalibration.rv.limits	scal-rvlimits	**-200.,200.**	Delta RV limits of the cross-correlation window in km/s
sgcalibration.rv.iterations	scal-rvniter	**3**	Maximum number of iterations used for the RV determination
sgcalibration.rv.wfactor	scal-rvwfactor	**1.5**	Data window width factor. The FWHM times this value determines the window width
sgcalibration.peak.iterations	scal-pfniter	**50**	Peak model fit maximum number of iterations
sgcalibration.peak.tests	scal-pfntest	**7**	Cross-correlation peak fit maximum number of tests
sgcalibration.peak.dchisquare	scal-pfdchisq	**0.0001**	Cross-correlation peak fit minimum chi-square difference

9.4.5 Product frames

Default file name	PRO.CATG	Description
arc_extspectra.fits	ARC_EXTSPECTRA	Extracted fluxes in ADU of the arc lamp spectra for each wavelength bin
arc_exterrors.fits	ARC_EXTERRORS	Statistical error (standard deviation) of the extracted arc lamp spectra fluxes for each wavelength bin
arc_extpixels.fits	ARC_EXTPIXELS	Number of pixels contributing to each wavelength bin of the extracted arc spectra (standard and Horne extraction only)
arc_extmodel.fits	ARC_EXTMODEL	The fibre profile model used to extract the observed fluxes. This frame is only available if the optimal extraction has been used. It is basically a simulation of the input frame and can be compared with that to assess the quality of the spectrum extraction.
arc_exttraces.fits	ARC_EXTTRACES	Centroid position in pixels of the extracted fluxes for each wavelength bin
dispersion_solution.fits	DISPERSION_SOLUTION	Computed optical model parameters and residuals correction
line_data.fits	LINE_DATA	Status information and line parameters for the last iteration of the wavelength calibration
slit_geometry_setup.fits	SLIT_GEOMETRY_SETUP	Table of corrected fibre positions in the focal plane
arc_rbnspectra.fits	ARC_RBNSPECTRA	Rebinned arc lamp spectra

Default file name	PRO.CATG	Description
arc_rbnerrors.fits	ARC_RBNERRORS	Statistical error (standard deviation) of the rebinned arc lamp spectra

The following products are stored using the extracted frame format (cf. section 6.1): ARC_EXTSPECTRA, ARC_EXTERRORS, ARC_EXTPIXELS, ARC_EXTTRACES, ARC_RBNSPECTRA, ARC_RBNERRORS.

DISPERSION_SOLUTION: The fitted optical model parameters are stored as the FITS header keywords `FCOLL`, `GCAM`, `THETA`, `SLITDX`, `SLITDY`, and `SLITPHI` of the keyword group `PRO.WSOL.OPTMOD`.

The table extension contains the fitted coefficients of two dimensional Chebyshev polynomial used to model the optical model residuals (see section 10.1.5.

LINE_DATA: Line status information for each line used during the last line search iteration. A detailed description is given in the following section.

SLIT_GEOMETRY_SETUP: See section 7.3.

ARC_RBNSPECTRA: The fluxes of the arc lamp spectra resampled to a regular wavelength grid. The wavelength is in units of nanometers.

The signal–to–noise ratio of the extracted and resampled spectra may simply be computed, dividing the resampled spectra by the error spectra of the ARC_RBNERRORS product.

ARC_RBNERRORS: The statistical error (standard deviation) of the resampled fluxes for each bin of the regular wavelength grid.

9.4.6 Quality control parameters

Quality control parameters are available for the rebinned arc lamp spectrum.

```
QC.OUT1.MEAN.RAW
QC.OUT1.NSAT.RAW
QC.LAMP.EFFIC
QC.LAMP.EFFIC1
QC.WSOL.REBIN.RMS
QC.RESOL.MEAN
QC.RESOL.RMS
QC.RESOL.NTOTAL
QC.RESOL.NLINES
QC.RESOL.POWER
```

For a detailed description of their definition see appedix B.

Status information on the computed dispersion solution can be found on the LINE_DATA ‘QC product’ generated by the recipe. The product contains a set of images used to store certain parameters for each line and fibre. In the first extension, the wavelength of the used arc lamp lines are stored. Each line is represented by one row in the status images. The wavelength increases from bottom to top. The second extensions contains the line status flags. Each line having a non–zero value has been rejected during the process.

The columns in the images correspond to the spectra as they are present in the extracted or rebinned arc lamp frame.

In the remaining extensions various line parameters are stored:

- Parameters of the line profile fit: amplitude, background, line center, line width (labelled Width1), line FWHM and possibly the exponent of the line profile function (labelled Width2).
- The uncertainties of the line parameters.
- Line positions on the detector (labelled Xccd and Yccd). Note that Xccd is the coordinate along the dispersion axis.
- line fit status information.

9.5 gistandard

The recipe processes a set of GIRAFFE standard star observations observed using the Argus or IFU instrument mode. If more than one standard star observation is provided as an input the observations are averaged. From this set of input frames the recipe creates a reduced standard star frame, and the corresponding extracted and rebinned spectra frames. If an extracted and normalised flat–field frame is provided, the extracted standard star frame can be corrected for the pixel–to–pixel variations and the fibre–to–fibre transmission using the parameter `flat-apply`. If only fibre–to–fibre transmission should be corrected for, the parameter `transmission-apply` may be used as an alternative. In this case the flat–field data is not used, but rather the transmission correction factors stored in the fibre setup table of the flat–field frame. Note that one should not use both options, `flat-apply` and `transmission-apply` at the same time.

The recipe also creates a data cube from the rebinned spectra, and an image of the reconstructed field(s) of view. The latter is obtained by integration of the data cube over its wavelength domain. In case of Argus observations the layout of the reconstructed field of view corresponds to the layout of the Argus microlens array, as it is shown in figure 7.2 (top panel) of [RD1]. In case of IFU observations the images of the individual field(s) of view of the used IFU buttons correspond to figure 2.7 of [RD1], and are arranged into a single image. While the reconstructed images are always created as a recipe product, the data cube is saved as a product file only if it is enabled using the option `recon-cube`. By default a data cube is created.

For description of the recipe algorithm and how it can be controlled using the parameters listed below, see section 10.2.5.

9.5.1 Input frames

The recipe expects one or more Argus or IFU standard star observations as an input. If more than one observations are present, they are averaged.

DO Category	Frame Type	Constraint	Parameter	Description
STD	raw	required		Raw spectrophotometric standard star observation (Argus and IFU mode only)

DO Category	Frame Type	Constraint	Parameter	Description
BAD_PIXEL_MAP	calibration	optional		Bad pixel map
MASTER_BIAS	calibration	optional	remove-bias	Master bias
MASTER_DARK	calibration	optional		Master dark
FF_LOCCENTROID, FF_PSFCENTROID	calibration	required		Fibre localisation, centroid position
FF_LOCWIDTH, FF_PSFWIDTH	calibration	required		Fibre localisation, spectrum width
FF_EXTSPECTRA	calibration	optional	flat-apply, transmission-apply	Extracted, normalised flat–field lamp spectra
FF_EXTERRORS	calibration	optional	flat-apply	Errors of the extracted, normalised flat–field lamp spectra
DISPERSION_SOLUTION	calibration	required		Dispersion solution
SLIT_GEOMETRY_SETUP	calibration	required		Setup specific slit geometry table
GRATING_DATA	static	required		Grating physical parameters
ATMOSPHERIC_EXTINCTION	static	required		Atmospheric extinction coeffi-cients table
FLUX_STANDARDS	static	required		Spectrophotometric standard star table

9.5.2 Recipe parameters

Parameter	Command line alias	Values (**default**, other)	Description
extraction.method	extr-method	**SUM**, HORNE, OPTIMAL	Spectrum extraction method
extraction.ron	extr-ron	**-1.**	New bias sigma (RON) value for bias and dark corrected image
extraction.psf.model	extr-psfmodel	psfexp, **psfexp2**	Analytical model of the fibre profile
extraction.psf.sigma	extr-psfsigma	**7.**	Sigma clipping threshold used for rejecting data points while fitting the fibre profile to the data.
extraction.psf.iterations	extr-psfniter	**2**	Maximum number of iterations used for fitting the fibre profile.
extraction.horne.extrawidth	extr-hewidth	**2**	Number of extra pixels added to the fibre half-width
extraction.horne.mingood	extr-hmingood	**3**	Minimum number of data points kept for fitting the fibre profile

Parameter	Command line alias	Values (**default**, other)	Description
extraction.optimal.fraction	extr-omfrac	**0.9**	Minimum fraction of data points kept for fitting the fibre profiles of a wavelength bin
extraction.optimal.wfactor	extr-owfactor	**3.**	Scale factor of the fibre half-width
extraction.optimal.bkgorder	extr-obkgorder	**2**	Order of the polynomial background model.
rebinning.xresiduals	rbin-xresid	**true**, false	Take optical model residuals into account during spectrum rebinning
rebinning.range	rbin-range	**setup**, common	Rebinning range
flat.apply	flat-apply	true, **false**	Turns the flat field correction on and off (implies fibre-to-fibre transmission correction)
transmission.apply	transmission-apply	true, **false**	Turns the fibre–to–fibre transmission correction on and off
fov.range.minimum	recon-min	**0.**	Minimum wavelength for image reconstruction
fov.range.maximum	recon-min	**0.**	Maximum wavelength for image reconstruction
fov.cube	recon-cube	**true**, false	Creates a data cube from the rebinned spectra as an additional product.
fov.cube.format	recon-format	**single**, eso3d	Selects the file format used for saving data cubes.
fxcalibration.sky.correct	flx-skycorr	true, **false**	Correct spectra for the sky emission

9.5.3 Detailed parameter description

rebinning.range Selects the wavelength range used for the spectrum rebinning. Setting the parameter to "setup" will use the wavelength range as it is defined in the grating data table for the instrument setup, whereas setting it to common will use the wavelength interval which is common to all allocated fibres.

flat.apply: Setting the parameter to "true" will apply the flat–field correction to the extracted standard star spectra. This also implies a fibre–to–fibre transmission correction. This option should not be used together with the option `transmission.apply`. If the errors of the extracted flat field (FF_EXTERRORS) are present in the set of frames, they are taken into account for the computation of the extracted spectra errors.

transmission.apply: Setting the parameter to "true" will use the transmission correction factors stored in the FIBER_SETUP table of the extracted flat–field frame to correct the extracted standard star spectra for differences in the fibre transmissivities. This option should not be used in conjunction with the option `flat.apply`.

fov.range.max: Sets the minimum wavelength of the wavelength interval, which is used for reconstructing the field(s) of view. The given wavelength must be in units of nanometers.

fov.range.max: Sets the maximum wavelength of the wavelength interval, which is used for reconstructing the field(s) of view. The given wavelength must be in units of nanometers.

fov.cube: The option controls whether the data cube(s) are saved as a pipeline product file or not.

fov.cube.format: The option controls the file format which is used for writing the data cube(s) to a file. By default the parameter is set to "single", which creates old-style data cube(s), i.e. data cube(s) and their error cube(s) are stored in separate files. Setting the parameter to "eso3d" will use the ESO3D format (cf. RD7) when writing the data cube(s). In this case both, the data and their errors are stored in a single file. Cubes stored in this format can be visualised using the ESO 3D Visualization tool (cf. RD8).

fxcalibration.sky.correct: Turns the correction for the sky emission on and off.

9.5.4 Extra recipe parameters

The following table summarises extra recipe parameters, which are not intended to be used for normal pipeline processing, but may be useful for expert users and pipeline maintainers.

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.remove	remove-bias	**true**, false	Turn bias removal on and off
biasremoval.method	bsremove-method	UNIFORM, PLANE, **PROFILE**, CURVE, MASTER, ZMASTER, PROFILE+CURVE, MASTER+PLANE, MASTER+CURVE, ZMASTER+PLANE, ZMASTER+CURVE	Method to use for correcting the bias.
biasremoval.areas	bsremove-areas	**5:40:0:4095**	Specification of image areas to be used for the bias correction
biasremoval.sigma	bsremove-sigma	**2.5**	Sigma clipping threshold factor
biasremoval.iterations	bsremove-niter	**5**	Sigma clipping number of iterations
biasremoval.fraction	bsremove-mfrac	**0.8**	Sigma clipping minimum fraction of accepted points
biasremoval.xorder	bsremove-xorder	**1**	Order of polynomial fit along X (method CURVE only)
biasremoval.yorder	bsremove-yorder	**1**	Order of polynomial fit along Y (method CURVE only)

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.xstep	bsremove-xstep	**1**	Sampling step along X (method CURVE only)
biasremoval.ystep	bsremove-ystep	**1**	Sampling step along Y (method CURVE only)
rebinning.method	rbin-method	**linear**, spline	Rebinning method
rebinning.scalemethod	rbin-scmethod	**linear**, log	Rebinning scaling method
rebinning.lstep	rbin-lstep	**0.005**	Lambda step size, only used if rebinning.scalemethod is "linear"
rebinning.size	rbin-size	**0**	Size of output rebinned spectra in terms of pixels. If it is set to 0 the size is computed from the wavelength range and lambda stepsize. If it is larger than 0 the lambda stepsize is adjusted so that the wavelength range is covered by the given number of pixels.

9.5.5 Product frames

Default file name	PRO.CATG	Description
std_reduced.fits	STD_REDUCED	Bias subtracted average of all input raw standard star observations
std_extspectra.fits	STD_EXTSPECTRA	The extracted fluxes in ADU of the standard star spectra for each wavelength bin. The fluxes are extracted from the reduced standard star frame. The spectra may also be flat field corrected.
std_exterrors.fits	STD_EXTERRORS	Statistical error (standard deviation) of the extracted standard star spectra
std_extpixels.fits	STD_EXTPIXELS	Number of pixels contributing to each extracted wavelength bin of the standard star spectra (standard and Horne extraction only)
std_extmodel.fits	STD_EXTMODEL	The fibre profile model used to extract the observed fluxes. This frame is only available if the optimal extraction has been used. It is basically a simulation of the input frame and can be compared with that to assess the quality of the spectrum extraction

Default file name	PRO.CATG	Description
std_exttraces.fits	STD_EXTTRACES	The centroid position of the extracted fluxes for each wavelength bin.
std_rbnspectra.fits	STD_RBNSPECTRA	Rebinned standard star spectra
std_rbnerrors.fits	STD_RBNERRORS	Statistical error (standard deviation) of the rebinned standard star spectra
std_rcspectra.fits	STD_RCSPECTRA	Reconstructed field of view
std_rcerrors.fits	STD_RCERRORS	Statistical error (standard deviation) of the reconstructed field of view
std_cube_spectra.fits	STD_CUBE_SPECTRA	Data cube of the rebinned spectra
std_cube_errors.fits	STD_CUBE_ERRORS	Data cube of the rebinned spectra errors
std_cube.fits	STD_CUBE	Data cube of the rebinned spectra and errors in ESO3D format
instrument_response.fits	INSTRUMENT_RESPONSE	Instrument response curve
efficiency_curve.fits	EFFICIENCY_CURVE	Instrument efficiency curve

The following products are stored using the extracted frame format (cf. section 6.1): STD_EXTSPECTRA, STD_EXTERRORS, STD_EXTPIXELS, STD_EXTTRACES, STD_RBNSPECTRA, STD_RBNERRORS.

STD_RBNSPECTRA: The fluxes of the standard star spectra resampled to a regular wavelength grid. The wavelength is in units of nanometers.

The signal–to–noise ratio of the extracted and resampled spectra may simply be computed, dividing the resampled spectra by the error spectra of the STD_RBNERRORS product.

STD_RBNERRORS: The statistical error (standard deviation) of the resampled fluxes for each bin of the wavelength grid. This statistical error contains the errors of the flat–field correction, if a flat–field correction was requested and an FF_EXTERRORS frame was provided. However, if only the fibre transmission correction was requested the contribution of the fibre transmission correction is not taken into account, but may be estimated from the column DTRANSMISSION of the FIBER_SETUP table.

STD_RCSPECTRA: This product is only present for IFU or Argus observations. Is an image of the field of view for each IFU head that was used during the observation. Therefore, for IFU observations more than one reconstructed field of view may be present. In this case each field of view is reconstructed from a single subslit (cf. column SSN in the fibre setup table of the resampled spectra product, for instance). They are arranged in the STD_RCSPECTRA product starting at the lower left corner and going from left to right and bottom to top. The image is generated by integrating the resampled fluxes over a common wavelength range. The pixel values are the total flux, integrated over the reconstruction wavelength range, of the fibre at this position in the IFU head. The association between the X and Y coordinates of a fibre in the reconstructed field of view and the corresponding spectrum is possible using the X and Y columns of the fibre setup table attached to, for instance, the resampled spectra product.

STD_RCERRORS: The product has the same structure as the STD_RCSPECTRA product. The pixel values are the statistical error (standard deviation) of the integrated fluxes of the fibre at that position in the IFU head. The image is the reconstructed from the STD_RBNERRORS product.

STD_CUBE_SPECTRA: In case of an Argus observation this product is a simple data cube stored in the primary data unit of the product FITS file, and in case of an IFU observation, the field-of-view of the individual IFU buttons are stored in separate FITS extensions (the primary data unit of the product file is empty). It is not stored in the Euro3D format. The spatial axis are the X and Y axis, and the wavelength axis is the Z axis of the cube. The product is created from the STD_RBN_SPECTRA frame.

The cube contains a simple world coordinate system which is constructed from the header and fiber setup information of the raw data. The world coordinate system specified in the FITS header is using the CD matrix convention and not the CDELT keywords to specify the axis scales. Note that this may give problems to certain visualisation tools.

STD_CUBE_ERRORS: The product has the same structure as the STD_CUBE_SPECTRA product. The pixel values are the statistical error (standard deviation) of the resampled fluxes. The product is created from the STD_RBNERRORS product.

STD_CUBE: The product contains the rebinned spectra and their associated error information in the ESO3D format (cf. RD7). If the ESO3D file format was selected as output format of the data cube(s), this product replaces the two recipe products STD_CUBE_SPECTRA and STD_CUBE_ERRORS, which will not be created in this case.

The world coordinates provided with this product are equivalent to the world coordinates provided by the STD_CUBE_SPECTRA product.

INSTRUMENT_RESPONSE : The product is the instrument response, defined as the factor by which the incident spectrum at the top of the atmosphere has to be multiplied to obtain the spectrum recorded on the detector. It is given in terms of $e^- cm^2 erg^{-1}$.

The response curve is stored as a 1-dimensional image in the primary data unit of the product fits file.

EFFICIENCY_CURVE: The product is the instrument efficiency, defined as the ratio of detected photons (corrected for the atmospheric extinction) over the number of incident photons.

The efficiency for each wavelength bin is stored in a FITS table.

9.5.6 Quality control parameters

Quality control parameters are available for the efficiency curve.

```
QC.OUT1.NSAT.RAW
QC.EFFICIENCY.MEAN
QC.SKY.MEAN
```

For a detailed description of their definition see appedix B.

9.6 giscience

The recipe processes a set of GIRAFFE science observations. If more than one science observations are specified, these are averaged. From this set of input frames the recipe creates a reduced science frame, and the extracted and rebinned spectra frames. If an extracted and normalised flat–field frame is present in the input, the

extracted science frame can be corrected for the pixel–to–pixel variations and the fibre–to–fibre transmission using the parameter `flat-apply`. If only fibre–to–fibre transmission should be corrected for, the parameter `transmission-apply` may be used as an alternative. In this case the flat–field data is not used, but rather the transmission correction factors stored in the fibre setup table of the flat–field frame. Note that one should not use both options, `flat-apply` and `transmission-apply` at the same time.

The observed spectra may be corrected for residual wavelength drifts using the simultaneous calibration fibres. The correction can be enabled by passing the option `siwc-apply` to the recipe. The wavelength drifts obtained from a cross-correlation with a reference mask of laboratory wavelength are then corrected. The correction applied to the wavelength calibrated spectra is written to the column WLRES of the FIBER_SETUP table. If necessary the correction can be undone by adding the values found in this column to the wavelengths associated with the pixels of the rebinned spectra frame.

If the observations have been done using the IFU or Argus fibre systems the recipe also creates a reconstructed image of the respective field(s) of view. The layout of the reconstructed field(s) of view corresponds to the layout of the Argus and IFU microlens arrays, as they are shown in figure 2.7 and figure 7.2 (top panel) of [RD1]. The reconstructed field(s) of view are generated from the created Argus and IFU data cubes. However, the data cubes are only saved as a product file if it has been enabled using the parameter `recon-cube`. By default a data cube is created.

For description of the recipe algorithm and how it can be controlled using the parameters listed below, see section 10.2.6.

9.6.1 Input frames

The recipe expects one or more science observation in input. If more than one are present, they are averaged.

DO Category	Frame Type	Constraint	Parameter	Description
SCIENCE	raw	required		Raw science observation
BAD_PIXEL_MAP	calibration	optional		Bad pixel map
MASTER_BIAS	calibration	optional	remove-bias	Master bias
MASTER_DARK	calibration	optional		Master dark
FF_LOCCENTROID, FF_PSFCENTROID	calibration	required		Fibre localisation, centroid position
FF_LOCWIDTH, FF_PSFWIDTH	calibration	required		Fibre localisation, spectrum width
FF_EXTSPECTRA	calibration	optional	flat-apply, transmission-apply	Extracted, normalised flat–field lamp spectra
FF_EXTERRORS	calibration	optional	flat-apply	Errors of the extracted, normalised flat–field lamp spectra

DO Category	Frame Type	Constraint	Parameter	Description
DISPERSION_SOLUTION	calibration	required		Dispersion solution
SLIT_GEOMETRY_SETUP	calibration	required		Setup specific slit geometry table
LINE_MASK	static	optional	siwc–apply	Line mask for this setup
GRATING_DATA	static	required		Grating physical parameters

9.6.2 Recipe parameters

Parameter	Command line alias	Values (**default**, other)	Description
extraction.method	extr-method	**SUM**, HORNE, OPTIMAL	Spectrum extraction method
extraction.ron	extr-ron	**-1.**	New bias sigma (RON) value for bias and dark corrected image
extraction.psf.model	extr-psfmodel	psfexp, **psfexp2**	Analytical model of the fibre profile
extraction.psf.sigma	extr-psfsigma	**7.**	Sigma clipping threshold used for rejecting data points while fitting the fibre profile to the data.
extraction.psf.iterations	extr-psfniter	**2**	Maximum number of iterations used for fitting the fibre profile.
extraction.horne.extrawidth	extr-hewidth	**2**	Number of extra pixels added to the fibre half-width
extraction.horne.mingood	extr-hmingood	**3**	Minimum number of data points kept for fitting the fibre profile
extraction.optimal.fraction	extr-omfrac	**0.9**	Minimum fraction of data points kept for fitting the fibre profiles of a wavelength bin
extraction.optimal.wfactor	extr-owfactor	**3.**	Scale factor of the fibre half-width
extraction.optimal.bkgorder	extr-obkgorder	**2**	Order of the polynomial background model.
rebinning.xresiduals	rbin-xresid	**true**, false	Take optical model residuals into account during spectrum rebinning
rebinning.range	rbin-range	**setup**, common	Rebinning range
flat.apply	flat-apply	**true**, false	Turns the flat field correction on and off (implies fibre-to-fibre transmission correction)

Parameter	Command line alias	Values (**default**, other)	Description
transmission.apply	transmission-apply	true, **false**	Turns the fibre–to–fibre transmission correction on and off
siwc.apply	siwc-apply	**true**, false	Enable simultaneous wavelength calibration correction
sgcalibration.zmax	scal-zmax	**10000.**	Maximum allowed pixel value. To be effective it must be larger than 0.
fov.range.minimum	recon-min	**0.**	Minimum wavelength for image reconstruction
fov.range.maximum	recon-min	**0.**	Maximum wavelength for image reconstruction
fov.cube	recon-cube	**true**, false	Creates a data cube from the rebinned spectra as an additional product.
fov.cube.format	recon-format	**single**, eso3d	Selects the file format used for saving data cubes.

9.6.3 Detailed parameter description

rebinning.range Selects the wavelength range used for the spectrum rebinning. Setting the parameter to "setup" will use the wavelength range as it is defined in the grating data table for the instrument setup, whereas setting it to common will use the wavelength interval which is common to all allocated fibres.

flat.apply: Setting the parameter to "true" will apply the flat–field correction to the extracted standard star spectra. This also implies a fibre–to–fibre transmission correction. This option should not be used together with the option `transmission.apply`. If the errors of the extracted flat field (FF_EXTERRORS) are present in the set of frames, they are taken into account for the computation of the extracted spectra errors.

transmission.apply: Setting the parameter to "true" will use the transmission correction factors stored in the FIBER_SETUP table of the extracted flat–field frame to correct the extracted standard star spectra for differences in the fibre transmissivities. This option should not be used in conjunction with the option `flat.apply`.

siwc.apply: If the simultaneous calibration lamps were used, this option enables the correction of the rebinned spectra for residual wavelength shifts. The wavelength shifts are obtained from a cross-correlation with a mask of laboratory wavelength, as it is used to generate slitgeometry tables. If the correction is used the wavelength mask for the cross-correlation (LINE_MASK) must be present in the input set of frames.

sgcalibration.zmax: This is a truncation parameter for the line intensities in order to avoid that the cross–correlation is affected by heavily different line intensities, i.e. the result is biased because of a dominating, bright line. To find the proper truncation value a few tries may be necessary. The truncation value is in units of ADU and is applied to the rebinned spectra.

fov.range.max: Sets the minimum wavelength of the wavelength interval, which is used for reconstructing the field(s) of view. The given wavelength must be in units of nanometers.

fov.range.max: Sets the maximum wavelength of the wavelength interval, which is used for reconstructing the field(s) of view. The given wavelength must be in units of nanometers.

fov.cube: The option controls whether the data cube(s) are saved as a pipeline product file or not.

fov.cube.format: The option controls the file format which is used for writing the data cube(s) to a file. By default the parameter is set to "single", which creates old-style data cube(s), i.e. data cube(s) and their error cube(s) are stored in separate files. Setting the parameter to "eso3d" will use the ESO3D format (cf. RD7) when writing the data cube(s). In this case both, the data and their errors are stored in a single file. Cubes stored in this format can be visualised using the ESO 3D Visualization tool (cf. RD8).

9.6.4 Extra recipe parameters

The following table summarises extra recipe parameters, which are not intended to be used for normal pipeline processing, but may be useful for expert users and pipeline maintainers.

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.remove	remove-bias	**true**, false	Turn bias removal on and off
biasremoval.method	bsremove-method	UNIFORM, PLANE, **PROFILE**, CURVE, MASTER, ZMASTER, PROFILE+CURVE, MASTER+PLANE, MASTER+CURVE, ZMASTER+PLANE, ZMASTER+CURVE	Method to use for correcting the bias.
biasremoval.areas	bsremove-areas	**5:40:0:4095**	Specification of image areas to be used for the bias correction
biasremoval.sigma	bsremove-sigma	**2.5**	Sigma clipping threshold factor
biasremoval.iterations	bsremove-niter	**5**	Sigma clipping number of iterations
biasremoval.fraction	bsremove-mfrac	**0.8**	Sigma clipping minimum fraction of accepted points
biasremoval.xorder	bsremove-xorder	**1**	Order of polynomial fit along X (method CURVE only)
biasremoval.yorder	bsremove-yorder	**1**	Order of polynomial fit along Y (method CURVE only)

Parameter	Command line alias	Values (**default**, other)	Description
biasremoval.xstep	bsremove-xstep	**1**	Sampling step along X (method CURVE only)
biasremoval.ystep	bsremove-ystep	**1**	Sampling step along Y (method CURVE only)
rebinning.method	rbin-method	**linear**, spline	Rebinning method
rebinning.scalemethod	rbin-scmethod	**linear**, log	Rebinning scaling method
rebinning.lstep	rbin-lstep	**0.005**	Lambda step size, only used if rebinning.scalemethod is "linear"
rebinning.size	rbin-size	**0**	Size of output rebinned spectra in terms of pixels. If it is set to 0 the size is computed from the wavelength range and lambda stepsize. If it is larger than 0 the lambda stepsize is adjusted so that the wavelength range is covered by the given number of pixels.
sgcalibration.iterations	scal-cniter	**1**	Maximum number of iterations for the cross-correlation to determine residual wavelength shifts
sgcalibration.cc.step	scal-cstep	**-0.005**	Cross-correlation step
sgcalibration.cc.domain	scal-cdomain	**0.,0.**	Restricts the cross-correlation to the given domain
sgcalibration.rv.limits	scal-rvlimits	**-200.,200.**	Delta RV limits of the cross-correlation window in km/s
sgcalibration.rv.iterations	scal-rvniter	**3**	Maximum number of iterations used for the RV determination
sgcalibration.rv.wfactor	scal-rvwfactor	**1.5**	Data window width factor. The FWHM times this value determines the window width
sgcalibration.peak.iterations	scal-pfniter	**50**	Peak model fit maximum number of iterations
sgcalibration.peak.tests	scal-pfntest	**7**	Cross-correlation peak fit maximum number of tests

Parameter	Command line alias	Values (**default**, other)	Description
sgcalibration.peak.dchisquare	scal-pfdchisq	**0.0001**	Cross-correlation peak fit minimum chi-square difference
sdp.format.generate	generate-SDP-format	true, **false**	Enables the creation of 1D spectra in the Science Data Product (SDP) format
sdp.nassoc.keys	dummy-association-keys	**0**	Indicates the number of dummy ASSO[N|C|M]n keywords to create in the SDP products

9.6.5 Product frames

Default file name	PRO.CATG	Description
science_reduced.fits	SCIENCE_REDUCED	Bias subtracted average of all input raw science observations
science_extspectra.fits	SCIENCE_EXTSPECTRA	The extracted fluxes in ADU of the science spectra for each wavelength bin. The fluxes are extracted from the reduced science frame. The spectra may also be flat field corrected.
science_exterrors.fits	SCIENCE_EXTERRORS	Statistical error (standard deviation) of the extracted science spectra
science_extpixels.fits	SCIENCE_EXTPIXELS	Number of pixels contributing to each extracted wavelength bin of the science spectra (standard and Horne extraction only)
science_extmodel.fits	SCIENCE_EXTMODEL	The fibre profile model used to extract the observed fluxes. This frame is only available if the optimal extraction has been used. It is basically a simulation of the input frame and can be compared with that to assess the quality of the spectrum extraction
science_exttraces.fits	SCIENCE_EXTTRACES	The centroid position of the extracted fluxes for each wavelength bin.
science_rbnspectra.fits	SCIENCE_RBNSPECTRA	Rebinned science spectra
science_rbnerrors.fits	SCIENCE_RBNERRORS	Statistical error (standard deviation) of the rebinned science spectra
science_rcspectra.fits	SCIENCE_RCSPECTRA	Reconstructed field of view
science_rcerrors.fits	SCIENCE_RCERRORS	Statistical error (standard deviation) of the reconstructed field of view
science_cube_spectra.fits	SCIENCE_CUBE_SPECTRA	Data cube of the rebinned spectra
science_cube_errors.fits	SCIENCE_CUBE_ERRORS	Data cube of the rebinned spectra errors

Default file name	PRO.CATG	Description
science_cube.fits	SCIENCE_CUBE	Data cube of the rebinned spectra and errors in ESO3D format
science_spectrum_$\langle n \rangle$.fits	SCIENCE_RBNSPEC_IDP	1D spectra in Science Data Product (SDP) format, with multiple files numbered (not necessary consecutively) by $\langle n \rangle$
science_ancillary.fits	ANCILLARY.MOSSKY	The ancillary file identical in structure to science_rbnspectra.fits but with the 1D SDP spectrum data stripped out

The following products are stored using the extracted frame format (cf. section 6.1): SCIENCE_EXTSPECTRA, SCIENCE_EXTERRORS, SCIENCE_EXTPIXELS, SCIENCE_EXTTRACES, SCIENCE_RBNSPECTRA, SCIENCE_RBNERRORS.

SCIENCE_RBNSPECTRA: The fluxes of the science spectra resampled to a regular wavelength grid. The wavelength is in units of nanometers.

The signal–to–noise ratio of the extracted and resampled spectra may simply be computed, dividing the resampled spectra by the error spectra of the SCIENCE_RBNERRORS product.

SCIENCE_RBNERRORS: The statistical error (standard deviation) of the resampled fluxes for each bin of the wavelength grid. This statistical error contains the errors of the flat–field correction, if a flat–field correction was requested and an FF_EXTERRORS frame was provided. However, if only the fibre transmission correction was requested the contribution of the fibre transmission correction is not taken into account, but may be estimated from the column DTRANSMISSION of the FIBER_SETUP table.

SCIENCE_RCSPECTRA: This product is only present for IFU or Argus observations. Is an image of the field of view for each IFU head that was used during the observation. Therefore, for IFU observations more than one reconstructed field of view may be present. In this case each field of view is reconstructed from a single subslit (cf. column SSN in the fibre setup table of the resampled spectra product, for instance). They are arranged in the SCIENCE_RCSPECTRA product starting at the lower left corner and going from left to right and bottom to top. The image is generated by integrating the resampled fluxes over a common wavelength range. The pixel values are the total flux, integrated over the reconstruction wavelength range, of the fibre at this position in the IFU head. The association between the X and Y coordinates of a fibre in the reconstructed field of view and the corresponding spectrum is possible using the X and Y columns of the fibre setup table attached to, for instance, the resampled spectra product.

SCIENCE_RCERRORS: The product has the same structure as the SCIENCE_RCSPECTRA product. The pixel values are the statistical error (standard deviation) of the integrated fluxes of the fibre at that position in the IFU head. The image is the reconstructed from the SCIENCE_RBNERRORS product.

SCIENCE_CUBE_SPECTRA: In case of an Argus observation this product is a simple data cube stored in the primary data unit of the product FITS file, and in case of an IFU observation, the field-of-view of the individual IFU buttons are stored in separate FITS extensions (the primary data unit of the product file is empty). It is not stored in the Euro3D format. The spatial axis are the X and Y axis, and the wavelength axis is the Z axis of the cube. The product is created from the SCIENCE_RBN_SPECTRA frame.

The cube contains a simple world coordinate system which is constructed from the header and fiber setup information of the raw data. The world coordinate system specified in the FITS header is using the

CD matrix convention and not the CDELT keywords to specify the axis scales. Note that this may give problems to certain visualisation tools.

SCIENCE_CUBE_ERRORS: The product has the same structure as the SCIENCE_CUBE_SPECTRA product. The pixel values are the statistical error (standard deviation) of the resampled fluxes. The product is created from the SCIENCE_RBNERRORS product.

SCIENCE_CUBE: The product contains the rebinned spectra and their associated error information in the ESO3D format (cf. RD7). If the ESO3D file format was selected as output format of the data cube(s), this product replaces the two recipe products SCIENCE_CUBE_SPECTRA and SCIENCE_CUBE_ERRORS, which will not be created in this case.

The world coordinates provided with this product are equivalent to the world coordinates provided by the SCIENCE_CUBE_SPECTRA product.

9.6.6 Quality control parameters

No quality control parameters are computed.

10 Algorithms

10.1 General Algorithms

10.1.1 Bias removal

The bias is either removed by means of a model which is computed on the pre and overscan regions of the raw frame, or by subtracting a master bias frame (method "MASTER"), possibly correcting for a bias drift (method "ZMASTER"). The model may simply be the mean value (method "UNIFORM") or a fitted surface (methods "PLANE" or "CURVE"). If a master bias frame is subtracted, the frame is subtracted as if there is no bad pixel information is available. If a bad pixel map is present, the bias value for the bad pixels is taken from the bias model. The default model used for the replacement of bad pixels is a fitted plane. This can be changed to a general 2-dimensional polynomial fit by adding "+CURVE" to the bias correction method strings "MASTER" or "ZMASTER" when the recipes are executed.

In the course of the upgrade of the GIRAFFE CCD it has been found that using a master flat field may lead to over corrections at the long wavelength end of the spectra. This is due to the fact that the shape of the first bias in a sequence differs slightly from the other ones, and only this first bias frame has the appropriate shape to correct any other single exposure. Therefore, the combination of N bias frames will result in a master bias with the wrong shape, since the shape is determined by the majority of the bias frames.

To avoid this over correction, a new bias correction method "PROFILE" was introduced, which uses the profile along the CCD columns (dispersion axis) of the given bias areas, computed directly on the raw frame to correct. The bias area used by default is the overscan area of the raw frame, which is free of artifacts which may be present in the prescan area at the beginning of the read out cycle[7]. The bias profile model is the mean profile computed from the CCD column profiles of each given bias area, where the individual CCD column profiles are a simple average of the CCD columns of that area.

However, this method may introduce low level line like features, which appear as straight lines in the extracted, and as curved lines in the wavelength calibrated spectum images. This is due to the fact that the same profile is subtracted from each CCD column, thereby replicating the features of subtracted profile across the CCD.

To avoid this propagation of the profile pattern the computed profile may be smoothed by fitting a one dimensional polynomial model to the profile data[8] The smoothing option is selected by adding the option "+CURVE" to the method string "PROFILE". The order of the smoothing polynomial is then taken from the recipe parameter specifying the y-order of the polynomial bias model. It has been found that 5 is a good choice for the order of the smoothing polynomial. If the profile smoothing is used, the error of the bias is determined as the error of the constant term of the fitting polynomial.

[7] Note that the physical overscan area corresponds to the first 50 pixels of raw frame!

[8] Similar results can be obtained by using the bias correction method "CURVE" on the overscan area with the polynomial order of 0 and 5 along x and y respectively. This may serve as a workaround, for older pipeline versions where the method "PROFILE" cannot be called with the option "CURVE".

10.1.2 Dark subtraction

Frames are corrected for the dark current by subtracting the scaled master dark frame from the input frame. In case of a bad pixel the mode of the master dark's pixel values is subtracted. If no bad pixel information is available, pixel values of the master dark smaller than a given threshold value are set to 0.

10.1.3 Fibre localisation

Standard localisation The position of the fibres is detemined using a flat–field frame. Each pixel row in the image is scanned from left to right and pixels exceeding a configurable threshold are searched:

$$P_{\text{limit}}(x,y) = \kappa\sqrt{\sigma_{\text{ron}}^2 + |P(x,y)|}$$

where $P(x,y)$ is the pixel value at the image coordinates x and y, σ_{ron} is the detector readout noise and κ is a user defined factor.

In order to be a spectrum pixel candidate, the value of a pixel must fulfil the relation:

$$P(x,y) - \overline{P}_{\text{bkg}} > P_{\text{limit}}(x,y)$$

where $\overline{P}_{\text{bkg}}$ is an estimate of the local background.

By scanning the image from left to right, the left border of a spectrum is marked by the first pixel exceeding the threshold. All following pixels exceeding the threshold value are part of that spectrum, and the last of these pixels defines the right spectrum border.

For each wavelength bin and each detected spectrum the barycenter of the pixel values within the spectrum borders is computed, defining the fibre centroid position. For each fiber the positions of the barycenter of all wavelength bins is then modeled using a Chebyshev polynomial.

The width of the detected fibres is a fit of a 2–dimensional model of the distance between the right and left spectrum border. As a model a 2–dimensional Chebyshev polynomial is used.

PSF localisation This localisation method uses an analytical model of the fibre PSF. Using the results from the standard localisation, in particular the determined fibre positions, the PSF model is fitted to the flat-field data for a number of wavelength bins. Combining several original flat-field pixels in a wavelength bin is used to improve the signal-to-noise ratio.

Three analytical PSF models are available. A Gaussian model, $P(x) = Ae^{-((x-x_0)/\omega)^2} + B$ and two exponential models $P(x) = Ae^{-(|x-x_0|^{\varepsilon}/\omega)} + B$, and $P(x) = Ae^{-(|x-x_0|/\omega)^{\varepsilon}} + B$ called *psfexp* and *psfexp2* respectively. The fit parameters of these models are the amplitude A, the background B, the position of the center x_0, the width ω and, if applicable, the exponent ε.

The profile parameters determined in that way are then modeled by either a 1 or 2-dimensional, low order Chebyshev polynomial, and the modeled profile parameters are stored to be used later, by the Horne and optimal extraction algorithms.

In addition to the profile center and the width ω of the PSF profile, the half-width at half maximum (HWHM) is computed and stored, so that the PSF localisation provides the same products as the standard localisation.

10.1.4 Spectrum extraction

The GIRAFFE pipeline provides 3 spectrum extraction methods.

Standard spectrum extraction The spectra are extracted by a simple summation along a virtual slit, that is defined by the fibre localisation solution derived from the flat–field. This method does not provide any correction for fibre cross-talk or a background subtraction.

Horne spectrum extraction Horne's method, for an a priori known profile is applied as described in [RD6]. The method applies also a background subtraction. The method works well for Medusa observations, where the fibres are well separated. For IFU and Argus observations, where fibre cross-talk is an issue, this method cannot be applied.

Optimal extraction This method extracts the fluxes of all spectra simultaneously by fitting the extraction model

$$\begin{aligned} m(x,y) &= B(x,y) + \sum_{j=0}^{N-1} f_{\mathrm{j}}(x) P_{\mathrm{j}}(x,y) \\ B(x,y) &= \sum_{k=0}^{l} a_{\mathrm{k}}(x) T_{\mathrm{k}}(y), \end{aligned}$$

to the observed data in each wavelength bin, where $f_{\mathrm{j}}(x)$ is the flux in the wavelength bin x originating from fibre j and $P_{\mathrm{j}}(x,y)$ is the fibre profile, which is assumed to be known. The background $B(x,y)$ is modeled by a Chebyshev polynomial of the order l, where a_{k} is the kth coefficient of the polynomial, and the T_{k} are the orthonormal basis functions.

The method treats the fiber cross-talk correctly and, at the same time, determines the background (i.e. the scattered light). A scattered light correction prior to the extraction of the spectra is not necessary. However, using a low order Chebyshev polynomial may not always be appropriate to model the background. In particular it may turn out, that the default 2nd-order polynomial is too rigid to properly represent the observed background.

The current implementation provides very good results for Medusa observations. For IFU and Argus observations the method is too slow to be used efficiently. Also, the implementation requires a known profile, which is taken from the PSF localisation. The way the profile parameters are determined in the PSF localisation does not take into account the effects of the fibre cross-talk, so that artefacts originating from the PSF localisation are propagated in the the extracted spectra, therby degrading the results.

Bad pixels are treated in the same way by all 3 methods. Bad pixels are skipped when the spectra are extracted and do not contribute to the extracted fluxes. In addition, the Horne and optimal extraction method reject data points of the observed signal s if their squared residuals, with respect to the model m, are larger than the expected variance v_{m} (computed from the model) times a given threshold factor σ_{th}, i.e. a data point s is ignored if the condition

$$(s(x,y) - m(x,y))^2 > \sigma_{\mathrm{th}} v_{\mathrm{m}}(x,y)$$

holds.

10.1.5 Wavelength calibration

The wavelength calibration in the GIRAFFE pipeline is based on fitting the GIRAFFE optical model:

$$x_{\rm opt}(\lambda,n) = \frac{Gf_{\rm coll}\left(\cos\vartheta\left(-\frac{\lambda m}{a}+\frac{X_{\rm f}(n)\cos\vartheta}{D}+\frac{f_{\rm coll}\sin\vartheta}{D}\right)+\sin\vartheta\sqrt{1-\left(\frac{Y_{\rm f}}{D}\right)^2-\left(-\frac{\lambda m}{a}+\frac{X_{\rm f}(n)\cos\vartheta}{D}+\frac{f_{\rm coll}\sin\vartheta}{D}\right)^2}\right)}{-\left(\sin\vartheta\left(-\frac{\lambda m}{a}+\frac{X_{\rm f}(n)\cos\vartheta}{D}+\frac{f_{\rm coll}\sin\vartheta}{D}\right)\right)+\cos\vartheta\sqrt{1-\left(\frac{Y_{\rm f}}{D}\right)^2-\left(-\frac{\lambda m}{a}+\frac{X_{\rm f}(n)\cos\vartheta}{D}+\frac{f_{\rm coll}\sin\vartheta}{D}\right)^2}}$$

where

$$\begin{aligned} X_{\rm f} &= x_{\rm f}\left(1+\varphi_{\rm s}y_{\rm f}\right)+x_{\rm s} \\ Y_{\rm f} &= y_{\rm f}\left(1-\varphi_{\rm s}^2\right)^{\frac{1}{2}}+y_{\rm s} \\ D &= \left(X_{\rm f}^2+Y_{\rm f}^2+f_{\rm coll}^2\right)^{\frac{1}{2}} \end{aligned}$$

and where G is the camera scale factor, $f_{\rm coll}$ the collimator focal length, ϑ the grating angle, a the grating groove spacing, and m the grating order. The variables $x_{\rm f}$ and $y_{\rm f}$ are the fibre positions in the focal plane, as given by the slit geometry table.

The variables $x_{\rm s}$, $y_{\rm s}$ and φ_s are the slit offsets and the slit rotation angle. The latter three are usually the only quantities which are adjusted when the optical model is fitted.

The residuals of the updated optical model with respect to the measured line positions are modelled using a Chebyshev polynomial model

$$x(\lambda,y)-x_{\rm opt}(\lambda,y)=\sum_j\sum_i c_{ij}T_i(\lambda)T_j(y)$$

where $T_n(x)$ are the Chebyshev polynomials.

The line positions on the detector are measured by fitting a line profile model to the pixel data within a window centered on the predicted line position.

In a last step, the slit geometry is updated. Imperfections in the slit assembly result in small offsets of the fibres with respect to each other. This effect is corrected, by adding small offsets to the fibre positions $x_{\rm f}$. The correction terms are computed by cross–correlating the rebinned arc spectrum, using the computed dispersion solution, with an appropriate line mask.

10.2 Recipe algorithms

10.2.1 gimasterbias

The raw bias frames are combined using the selected frame stacking method (recipe parameter `stack-method`). The resulting image is the master bias. If the generation of the bad pixel map was requested, all pixels $P(x,y)$ of the master bias frame which violates the condition

$$M_{\rm MB}-\kappa\sigma_{\rm ron}\le P(x,y)\le M_{\rm MB}+\kappa\sigma_{\rm ron}$$

where $M_{\rm MB}$ is the median pixel value of the master bias frame, $\sigma_{\rm ron}$ is the CCD readout noise, and t is the threshold factor, are flagged in the bad pixel map as bad pixels. The threshold factor κ may be adjusted using the parameter `bpm-factor` (see section 9).

10.2.2 gimasterdark

Each raw dark frame present in the input set of frames is corrected for the bias. Bad pixels are taken into account if a bad pixel mask is present in the set of frames. Each dark frame is then scaled to an exposure time of 1 second and the scaled frames are combined using the selected frame stacking method (recipe parameter `stack-method`). If a bad bixel mask is present, the values of the corresponding pixels in the combined image are set to 0. The resulting image is the master dark.

10.2.3 gimasterflat

The recipe avarages all fibre flat–field frames which are present in the input data set and the bias is removed from the averaged flat field lamp exposure. The resulting image is the master flat field frame. The master flat field frame is scanned from left to right for pixels exceeding an adjustable threshold (recipe parameter `sloc-noise`. Polynomial models of the fibre centroid and the fibre width are constructed using Chebyshev polynomials. The order of the polynomial models may be set using the parameter `sloc-yorder` for the model of the fibre centroid and `sloc-worder` for the model of the fibre width. The detected spectra are identified with the fibres from the setup found either in an already existing localisation solution, used as reference, or from the fibre setup specified on the command line, using the parameters `fiber-splist` or `fiber-nspec`.

Starting from the determined fibre centroids and width the localisation step is then repeated, this time a fit of an analytical PSF model is fitted to the master flat field data. The model can be chosen with the parameter `psf-model`. The fitted parameters of the fibre PSF are then modeled by a 1-dimensional (for each fibre), or a 2-dimensional (parameter `psf-prmfit`) polynomial model. The modeled parameters are stored for being used by the spectrum extraction. The order of the Chebyshev polynomial fit of the profile parameters can be changed using the options `psf-yorder` and `psf-worder`. Based on the modeled parameters the localisation solution, fibre centroids and width, is generated.

Using the computed fibre localisation, the flat–field spectra are extracted and normalised to the mean intensity of the extracted flat–field. The applied scaling factor is available from the FITS header of the product file. The standard error of the the extracted spectra, which is also provided, is treated in the same way.

Optionally, a scattered light model and the fiber–to–fibre transmission factors are computed. The scattered light model is a 2-dimensional, low order Chebyshev polynomial model of the inter-spectra regions of the master flat field frame. Therefore, a reliable model may only be obtained for Medusa observations, due to the limited number of intra-spectra pixels for Argus and IFU observations. At least, a scattered light model obtained from an Argus or an IFU observation needs to be carefully checked.

The fibre–to–fibre transmission factors are obtained by rebinning the flat-field spectra and comparing the flux integrated over a common wavelength interval of individual fibres to the reference fibre. As reference fibre the brightest fibre found in the master flat field frame is used. The transmission factors are computed for all, but the the simultaneous wavelength calibration fibres.

10.2.4 giwavecalibration

The recipe extracts the spectra from the bias corrected arc lamp frame, using the provided fibre localisation.

From the line catalog a set of calibration lines is selected, taking the wavelength range of the current instrument setup and the line quality flags (for instance blended or saturated) specified in the catalog into account. In addition the set of calibration lines is cleaned by the recipe using criteria for the line brightness, the line separation and for the relative flux ratio.

Using the cleaned line list and the slit geometry for the current instrument setup the dispersion solution is computed by fitting the GIRAFFE optical model to the detected line positions on the detector. The residuals of the fitted optical model with respect to the detected line positions are modeled, in a second step, by a Chebyshev polynomial. Both, the optical model and the Chebyshev model of the optical model residuals define the dispersion solution.

The input slit geometry can be updated with respect to the new dispersion solution using the recipe parameter `wcal.slit`. This step requires several iterations of adjusting the slit geometry (recipe option `scal-niter`) by cross-correlating the resampled arc lamp frame with a mask for selected calibration lines.

Finally, as an option, the input arc lamp frame is resampled to a regular wavelength grid using the computed dispersion solution (recipe parameter `wcal.rebin`), and the, possibly updated, slit geometry.

10.2.5 gistandard

If more than one raw standard star observation is provided, the raw observations are initially averaged, and the averaged standard star observations is used as input for the following processing steps. In the next step the bias is removed and, optionally the dark, if it has been provided. Both corrections take a bad pixel map into account if it is present.

In the following steps the spectra are extracted and wavelength calibrated. Note that, by default, the extracted spectra are neither flat-field corrected, nor corrected for the fibre–to–fibre transmission.

At this stage, a data cube and a reconstruction of the field(s) of view is created. Whether the data cube is saved as a product can be chosen using the option `recon-cube`. By default, the cube product is saved.

Finally the instrument response and the efficiency are computed by comparison with a reference standard star spectrum. For this comparison, the flux measured in all fibres is used. There is no source detection taking place on the reconstructed field of view images. Also the sky background is, by default, not subtracted, however this can be configured using the recipe option `flx-skycorr`. However, this sky correction simply subtracts the median sky level (for each wavelength bin) computed from the available sky fibres. The storage format and the units of the created instrument response are such it could directly be applied to a rebinned spectrum image, whereas the efficiency is the number of detected photons (corrected for atmospheric extinction) over the number of incident photons, tabulated for each wavelength bin and serves only instrument monitoring purposes.

10.2.6 giscience

At first, if more than one science observation is present in the input data set, all raw science frames are averaged. The averaged science frame is corrected for the bias and, optionally, for the dark current. The latter correction is only applied if a master dark frame is present in the input set of frames. The result of this is the reduced science observation, which is stored as a recipe product file.

Using the reduced science observation, the spectra are extracted and, optionally, corrected for fringing, pixel–to–pixel variations and the fibre–to–fibre transmission, by applying a flat–field correction (recipe parameter `flat-apply`). Alternatively one may correct only for the fibre–to–fibre transmission by scaling the detected flux of each fibre using the option `transmission-apply` instead. Eventually, the extracted spectra are wavelength calibrated and resampled to a wavelength grid.

If the simultaneous calibration lamps were used during the observation, and if the set of frames contains a line mask for the cross-correlation, the rebinned spectra are corrected for a residual wavelength drift. The residual shifts are obtained by cross-correlating the arc-lamp spectra of the simultaneous calibration fibres with the line mask. The wavelength offset obtained from the cross-correlation are then linearly interpolated across the CCD (spatial direction), and the interpolated offsets are applied to each fibre. The correction may be disabled using the option `siwc-apply`.

In addition, the recipe computes for each fibre the barycentric, heliocentric and geocentric corrections. However, this correction is not applied, but only stored in the fiber setup table of the rebinned spectra products.

For Argus and IFU observations data cubes are created. A single one in the case of an Argus observation and, in the case of an IFU observation, one for each IFU button. By default the data cube(s) are saved as a recipe product file. If this is not desired, it may be disabled with the recipe option `recon-cube`. For IFU observations the data cube of each IFU button is stored in a separate FITS extension.

Finally, the field(s) of view are reconstructed from the data cube using a common wavelength range of the calibrated spectra. The default wavelength range, computed from the instrument setup, may be adjusted using the recipe parameters `recon-min` and `recon-max`. The resulting image(s) are saved as a recipe product file.

A Installation

This chapter gives generic instructions on how to obtain, build and install the GIRAFFE pipeline. Even if this chapter is kept as up-to-date as much as possible, it may not be fully applicable to a particular release. This might especially happen for patch releases. One is therefore advised to read the installation instructions delivered with the GIRAFFE pipeline distribution kit. These release-specific instructions can be found in the file `README` located in the top-level directory of the unpacked GIRAFFE pipeline source tree. The supported platforms are listed in section A.1. It is recommended reading through section A.2.3 before starting the installation.

ESO pipelines can be installed via several methods, depending on your OS, most of which facilitate easy installation, upgrade and removal. Please see the *ESO Data Reduction Pipelines and Workflow Systems* page (*https://www.eso.org/pipelines*).

A.1 Supported platforms

The GIRAFFE pipeline is verified on the the VLT target platform:

- CentOS 7.3 (x86_64), using gcc 4.8.5,

In addition, the GIRAFFE pipeline has been built successfully on the following platforms (x86_64) using the default system compiler:

- Linux Fedora 27
- Linux Fedora 28
- Linux Fedora 29
- Linux Fedora 30
- Linux Fedora 31
- Linux openSUSE Leap 42.1
- Linux openSUSE Leap 42.2
- Linux openSUSE Leap 42.3
- Linux openSUSE Leap 15.0
- Linux openSUSE Leap 15.1
- Linux Debian 9
- Linux Ubuntu 16.04 LTS
- Linux Scientific Linux 7
- Linux CentOS 7

- macOS 10.11
- macOS 10.12
- macOS 10.13
- macOS 10.14

A.2 Building the GIRAFFE pipeline

This section shows how to obtain, build and install the GIRAFFE pipeline from the official source distribution.

A.2.1 Requirements

To compile and install the GIRAFFE pipeline one needs:

- an appropriate version of the GNU C/C++ compiler,
- a version of the `tar` file-archiving program,
- the GNU `gzip` and `make` utilities, and
- a `perl` installation.

Using Gasgano (included in the distribution) to launch pipeline recipes the CPL interfaces for Gasgano must be available. Building the CPL interfaces on an installation of the Java Development Kit (JDK 6 - JDK 9) is required.

To run Gasgano as a stand alone data browser the Java Runtime Environment (JRE 6, or newer) is sufficient.

A.2.2 Downloading the GIRAFFE pipeline distribution

The latest release of the GIRAFFE pipeline is available at `http://www.eso.org/sci/software/pipelines`, as a compressed tar–archive.
The distribution file is named like `giraf-kit-<x.y.z>.tar.gz`, where `<x.y.z>` is the placeholder for the actual package version number.

A.2.3 Compiling and installing the GIRAFFE pipeline

It is recommended to read through this section before starting the installation. Please note that in the following examples the placeholder of the package version `<x.y.z>` needs to be replaced by the actual package version string.

1. Unpack the GIRAFFE pipeline distribution using the following command:

```
tar -zxvf giraf-kit-<x.y.z>.tar.gz
```

at the shell prompt. This will create a directory `giraf-kit-<x.y.z>` containing the pipeline subpackages.

Note that to keep the size of the unpacked GIRAFFE pipeline distribution small the standard calibration data sets are distributed separately. They can be obtained from http://www.eso.org/sci/software/pipelines. The required disk space for a complete installation of the pipeline, including all standard calibrations, is about 2.6 GB.

2. Change to the directory `giraf-kit-<x.y.z>`.
3. Execute the pipeline installer[9], using:

```
./install_pipeline
```

The installer will then ask for the two installation directories for the software components and the calibration data respectively. If the script's defaults for the target directories are acceptable, they can be comfirmed by pressing <Enter>. Otherwise an appropriate path should be given.

4. The installer will install all required pipeline components. After the installation was completed successfully a list of available pipeline recipes is shown.

After the installation is finished, the directory tree containig the unpacked pipeline distribution is no longer needed and can be removed to save disk space.

A.2.4 Configuring the pipeline recipe front–end applications

For detailed information on how to setup the front–end applications EsoRexand Gasgano, please refer to their documentation, which is available at http://www.eso.org/sci/software/cpl/esorex.html and http://www.eso.org/sci/software/gasgano respectively.

[9] An installation of the pipeline without using the installation script is always possible, but is only recommended for an experienced user.

B Quality control parameter definitions

In the following the definitions of the quality control parameters are listed, as they are computed by the various GIRAFFE pipeline recipes:

```
Parameter Name:    QC DID
Class:             header|qc-log
Context:           process
Type:              string
Value Format:      %30s
Unit:
Comment Field:     Data dictionary for QC
Description:       Name/version of ESO DID to which QC keywords comply.

Parameter Name:    QC BIAS MASTER MEDIAN
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     Median master bias level (ADU)
Description:       Median value of all pixel values within the range
                   [100, 300[ ADU of the product master bias.

Parameter Name:    QC BIAS MASTER MEAN
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     Mean master bias level (ADU)
Description:       Mean value of all pixel values within the range
                   [100, 300[ ADU of the product master bias.

Parameter Name:    QC BIAS MASTER RMS
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     RMS of master bias level (ADU)
Description:       Population standard deviation of all pixel values
                   within the range [100, 300[ ADU of the product master bias.

Parameter Name:    QC OUT1 RON RAW
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     Read out noise of raw bias
```

```
Description:        Median of RMS values computed in 5 sub-windows (100x100
                    pixels) of the difference image of the first and second
                    raw bias frames in the input set of frames, devided by
                    the square root of 2.

Parameter Name:     QC OUT1 RON MASTER
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               ADU
Comment Field:      Read out noise of master bias
Description:        Median of RMS values computed in 5 sub-windows (100x100
                    pixels) of the master bias frame for pixel values in the
                    range [100,300[.

Parameter Name:     QC OUT1 STRUCT X
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               ADU
Comment Field:      Structure along the x axis.
Description:        The RMS of pixel values of the master bias frame (averaged
                    over its rows) is computed with respect to their mean
                    value. Pixel values outside the interval ]mean-2, mean+2[
                    are ignored.

Parameter Name:     QC OUT1 STRUCT Y
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               ADU
Comment Field:      Structure along the y axis.
Description:        The RMS of pixel values of the master bias frame (summed
                    over its rows) is computed with respect to their mean
                    value. Pixel values outside the interval ]mean-2, mean+2[
                    are ignored.

Parameter Name:     QC OUT1 MEAN RAW
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               ADU
Comment Field:      Mean level of first raw frame
Description:        Mean level of the first raw frame. The mean level
                    is computed on the whole raw image excluding the pre- and
                    overscan areas.
```

```
Parameter Name:    QC OUT1 NSAT RAW
Class:             header|qc-log
Context:           process
Type:              integer
Value Format:      %d
Unit:              ADU
Comment Field:     Number of saturated pixels in the first raw frame.
Description:       The number of pixels with a pixel value larger than
                   60000.

Parameter Name:    QC DARK CURRENT
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU/hr
Comment Field:     Dark current in ADU per hour.
Description:       Average pixel value of the bias subtracted dark frame,
                   computed across the whole chip or a central window, and
                                  divided by the exposure time in hours.

Parameter Name:    QC GLOW LEVEL
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU/s
Comment Field:     Total flux of glow feature.
Description:       This is the integrated flux of the masterdark frame in
                   the window [1350,3800:2048,4095].

Parameter Name:    QC GLOW POSX
Class:             header|qc-log
Context:           process
Type:              int
Value Format:      %d
Unit:              pixel
Comment Field:     X position of the glow feature.
Description:       X position of the peak flux in the window
                   [1350,3800:2048,4095].

Parameter Name:    QC GLOW POSY
Class:             header|qc-log
Context:           process
Type:              int
Value Format:      %d
Unit:              pixel
Comment Field:     Y position of the glow feature.
Description:       Y position of the peak flux in the window
                   [1350,3800:2048,4095].
```

```
Parameter Name:     QC LAMP EFFIC
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               ADU/sec
Comment Field:      Calibration lamp efficiency
Description:        Measures the lamp efficiency. It is the average flux
                    level of the extracted spectra, excluding the SIMCAL fibers
                    devided by the exposure time.

Parameter Name:     QC LAMP EFFIC1
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               ADU/sec
Comment Field:      SIMCAL lamp efficiency
Description:        Measures the lamp efficiency of the simultaneous
                    calibration lamp. It is the average flux level of the
                    extracted SIMCAL spectra devided by the exposure time.

Parameter Name:     QC FIBRE WIDTH MEAN
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               pixel
Comment Field:      Mean fibre half width
Description:        Mean of the mean fibre half widths.

Parameter Name:     QC FIBRE WIDTH RMS
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               pixel
Comment Field:      RMS of fibre half width
Description:        Mean RMS of the fibre half widths.

Parameter Name:     QC FIBRE CENTROID RMS
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               pixel
Comment Field:      RMS of fibre half width
Description:        The mean RMS of the fibre positions along the slit.

Parameter Name:     QC FIBRE CENTROID DIFF
Class:              header|qc-log
```

```
Context:            process
Type:               double
Value Format:       %e
Unit:               pixel
Comment Field:      Mean difference of fibre curvature.
Description:        The minimum and maximum position along the slit is measured
                    for each fibre. The differences, maximum position minus
                    minimum position, are averaged.

Parameter Name:     QC FIBRE TRANS MEDIAN
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:
Comment Field:      Median of the relative fibre transmission.
Description:        The median of the relative fibre transmission is computed
                    for fibres with a relative transmission within the
                    interval ]0.5, 2.0[. The simultaneous calibration fibres
                    are ignored.

Parameter Name:     QC FIBRE TRANS RMS
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:
Comment Field:      RMS of the relative fibre transmission.
Description:        The RMS of the relative fibre transmission is computed
                    for fibres with a relative transmission within the
                    interval ]0.5, 2.0[. The simultaneous calibration fibres
                    are ignored.

Parameter Name:     QC FIBRE TRANS ERROR
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:
Comment Field:      Median of the relative fibre transmission errors.
Description:        The median of the relative fibre transmission errors is
                    computed for fibres with a relative transmission within
                    the interval ]0.5, 2.0[. The simultaneous calibration
                    fibres are ignored.

Parameter Name:     QC WSOL REBIN RMS
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:
```

```
Comment Field:      Straightness of rebinned emission lines.
Description:        Using the logarithm of the rebinned spectrum image,
                    the difference of the fluxes of spectrum number 3
                    and the average spectrum, obtained by averaging all
                    spectra excluding the simultaneous calibration
                    fibres along the cross-dispersion direction is
                    computed. The difference image is converted to a
                    linear scale and the RMS of the flux difference is
                    computed.

Parameter Name:     QC RESOL MEAN
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               nm
Comment Field:      Average line FWHM.
Description:        Initially the mean line FWHM [pxl] and the standard
                    deviation is computed ignoring all lines with an FWHM
                    larger than 100. The mean line FWHM is then re-computed
                    ignoring all lines with an FWHM outside of the 3 sigma
                    interval around the mean.The computed mean is converted
                    from pixels to nm using a linearized dispersion relation.

Parameter Name:     QC RESOL RMS
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               nm
Comment Field:      RMS of line FWHM.
Description:        The RMS of the line FWHM [pxl] is computed ignoring all
                    values exceeding the 10 sigma rejection threshold (cf.
                    QC.RESOL.MEAN). The computed RMS is converted from pixels
                    to nm using a linearized dispersion relation.

Parameter Name:     QC RESOL NTOTAL
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %d
Unit:
Comment Field:      Total number of spectral lines.
Description:        The total number of available spectral lines.

Parameter Name:     QC RESOL NLINES
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %d
Unit:
```

```
Comment Field:      Number of spectral lines used.
Description:        The number of spectral lines which were actually used to
                    compute the resolving power, after applying the filter
                    criterion.

Parameter Name:     QC RESOL POWER
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:
Comment Field:      Resolving power.
Description:        The central wavelength of the grating divided by the
                    average line FWHM computed from a linearized dispersion
                    relation.

Parameter Name:     QC EFFICIENCY MEAN
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:
Comment Field:      Mean spectrograph efficiency
Description:        From the efficiency curve an average efficiency value is
                    computed on an interval of central wavelength +/- 10 percent
                    of the full spectral range of the rebinned spectrum.

Parameter Name:     QC SKY MEAN
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               ADU
Comment Field:      Mean sky level
Description:        Average level of the sky spectrum.
```