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

FLAMES-UVES Pipeline User Manual - Draft not for release

VLT-MAN-ESO-19500-3016

Issue 5.0

Date 2008-04-02

Prepared:	Andrea Modigliani and Jonas Møller Larsen	2008-04-01		
	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
5.0	21/04/2008		Draft describing release 4.2.2

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

The FLAMES-UVES pipeline is a subsystem of the *VLT Data Flow System* (DFS). Its target user is ESO *Data Flow Operations* (DFO) in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. It should also serve as a quick look tool for *Paranal Science Operations* (PSO). Additionally, the FLAMES-UVES 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 FLAMES-UVES data reduction sequence with the FLAMES-UVES pipeline.

This manual is a complete description of the data reduction recipes implemented by the CPL based FLAMES-UVES pipeline, reflecting the status of the FLAMES-UVES pipeline as of Apr 21, 2008 (version 4.2.2). Release 4.2.2 supports the reduction of UVES frames obtained when the UVES slit is fed by the fiber link to FLAMES.

1.2 Acknowledgements

The FLAMES-UVES pipeline has been initially developed as a MIDAS package by Andrea Modigliani, as a wrapper of the FLAMES-UVES Data Reduction Software developed by Giacomo Mulas, Ignazio Porceddu and Francesco Damiani of the Ital-FLAMES consortium. Then to uniform implementation across pipelines the pipeline has been ported to CPL by Andrea Modigliani and Jonas Møller Larsen. Jonas implemented the low level CPL wrappers to the MIDAS environment C interface functions, converted the uves_cal_prep_sff_ofpos and uves_cal_wavecal recipes, part of the reduction chain recipe and wrote scripts to automatically test the recipes and run regression tests to the MIDAS recipes. Andrea extended the master flat, single fibre order trace and physical model recipes to support frames used in the FIBER mode, and fully ported the science recipe and completed the reduction chain recipe, added the support of cube output to the uves_cal_prep_sff_ofpos recipe and completed the testing of the whole package.

1.3 Scope

This document describes the CPL based FLAMES-UVES pipeline used at ESO-Garching and ESO-Paranal for the purpose of data assessment and data quality control.

Updated versions of the present document may be found on [1]. For general information about the current instrument pipelines status we remind the user of [2]. Quality control information are at [3].

Additional information on QFITS, the Common Pipeline Library (CPL) and EsoRex can be found respectively at [4], [5], [6]. The Gasgano front end is described in [7]. A description of the instrument is in [8,9,10]. The FLAMES-UVES instrument user manual is in [10]. The FLAMES-UVES calibration plan is in [11]. while results of Science Verifications (SV) are at [12]. Additional information on the DFS and VLT data interfaces are in [13,14] and [15]. A clear and compact description of the FLAMES-UVES pipeline is in [16].

1.4 Reference documents

[1]	UVES Pipeline Users' Manual	VLT-MAN-ESO-19500-2965 www.eso.org/pipelines
[2]	Current pipeline status	www.eso.org/observing/dfo/quality/pipeline-status.html
[3]	ESO-Data Flow Operation home page	www.eso.org/observing/dfo/quality/
[4]	QFITS home page	www.eso.org/projects/aot/qfits/
[5]	CPL home page	www.eso.org/cpl
[6]	EsoRex home page	www.eso.org/cpl/esorex.html
[7]	Gasgano User's Manual	VLT-PRO-ESO-19000-1932
[8]	UVES home page	www.eso.org/instruments/flames
[9]	*Installation and commissioning of FLAMES, the VLT Multifibre Facility*	The Messenger, **110**, 1, 2002.
[10]	VLT Paranal Science Operations FLAMES User Manual	VLT-MAN-ESO-13700-2994 www.eso.org/instruments/flames/doc
[12]	FLAMES SV home page	www.eso.org/sci/activities/vltsv/flamessv
[13]	VLT Data Flow System Specifications for Pipeline and Quality Control	VLT-SPE-ESO-19600-1233
[14]	DFS Pipeline & Quality Control – User Manual	VLT-MAN-ESO-19500-1619
[15]	ESO DICB – Data Interface Control Document	GEN-SPE-ESO-00000-0794
[16]	FLAMES-UVES Data Reduction Software Design and Architecture Report	VLT-TRE-ITA-13750-0002 www.eso.org/org/dmd/DFS-related-papers.html
[17]	Automatic data reduction in support of the FLAMES-UVES VLT facility Proceedings of the SPIE, Volume 4844, pp. 310-320 (2002).	www.eso.org/org/dmd/DFS-related-papers.html
[18]	UVES User Manual	VLT-MAN-ESO-13200-1825 www.eso.org/instruments/uves/doc
[19]	The FLAMES-UVES Pipeline	The ESO Messenger, 118, 8.

2 Overview

In collaboration with instrument consortia, the Pipeline Systems Department (PSD) of the Software Development 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 Gasgano.

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* [14] and *EsoRex*, both included in the pipeline distribution (see Appendix A, page 78). These applications can also be downloaded separately from `www.eso.org/gasgano` and `www.eso.org/cpl/esorex.html`. An illustrated introduction to Gasgano is provided in Section 5.

The FLAMES facility and the different types of FLAMES-UVES raw frames and auxilliary data are described in Sections 4, 7, and 8.

A brief introduction to the usage of the available reduction recipes using Gasgano or EsoRex is presented in Section 5. In section 6 we advice the user about known data reduction problems.

An overview of the data reduction, the input data, and the recipes involved in the calibration cascade is provided in section 9.

More details on inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in section 10.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 11.

In Appendix A the installation of the FLAMES-UVES pipeline recipes is described, in Appendix B the full list of changes is given, and in Appendix C a list of used abbreviations and acronyms is given.

3 What's new in pipeline release 4.2.2

The following major changes has been realised on the FLAMES-UVES pipeline:

- The pipeline has been ported to CPL.

For a more detailed list of changes see Appendix B.

4 FLAMES Instrument Description

FLAMES is the multi-object, intermediate and high resolution spectrograph of the VLT. Mounted at UT2, FLAMES can access targets over a field of view 25 arcmin in diameter. FLAMES feeds two different spectrograph covering the whole visual spectral range: GIRAFFE and UVES. GIRAFFE allows the observation of up to 130 targets at the time or to do integral field spectroscopy, with intermediate resolution (either R 25000 or R 10000). UVES provides the maximum possible resolution (R=47000) but can access only up to 8 objects at the time. The instrument has been made available to the community and started operations in Paranal on April 1^{st}, 2000.

In this chapter a brief description of the FLAMES-UVES fiber link and the UVES instrument is given. A more complete documentation can be found in the FLAMES and UVES User Manuals [10,18].

Figure 4.0.1: A photo of FLAMES mounted at the Nasmyth A focus of Kuyen (VLT-UT2).

4.1 Instrument overview

FLAMES is the multi-object, intermediate and high resolution spectrograph of the VLT. Mounted at the Nasmyth A platform of UT2, FLAMES can access targets over a large corrected field of view (25 arcmin diameter). It consists of three main components:

- A Fibre Positioner (OzPoz) hosting two plates: while one plate is observing the other positions the fibres for the subsequent observations, therefore limiting the dead time between one observation and the next to less than 15 minutes, including the telescope preset and the acquisition of the next field.

- A medium-high resolution optical spectrograph, GIRAFFE, with three types of feeding fibre systems : MEDUSA, IFU, ARGUS.
- A link to the UVES spectrograph (Red Arm) via 8 single fibres of 1 arcsec entrance aperture.

Special observing software (FLAMES OS) coordinates the operation of the different subsystems, also allowing simultaneous acquisition of UVES and GIRAFFE observations with the observing modes listed in Table below. For combined observations, the exposure times for UVES and GIRAFFE do not need to be the same. Note that it is not possible to observe simultaneously in two GIRAFFE modes, or to observe the same target simultaneously with the two spectrographs.

UVES is the high resolution spectrograph at UT2 of the VLT (see Section 6.4). It was designed to work in long slit mode but it has been possible to add a fibre mode (6 to 8 fibres, depending on setup and/or mode) fed by the FLAMES positioner to its Red Arm only. Only the three standard UVES Red setups are offered, with central wavelength of 520, 580 and 860 nm respectively (see the manual for details). The standard readout mode of FLAMES-UVES is 225 kHz (unbinned) which ensures low readout noise. As of P76 a high-speed readout mode (625 kHz, unbinned, low gain) with increased readout noise but less overheads is offered in visitor mode only. No pipeline support is available in this mode. With an aperture on the sky of 1 arcsec, the fibres project onto 5 UVES pixels giving a resolving power of 47000. For faint objects and depending on the spectral region, one or more fibres can be devoted to recording the sky contribution. In addition, for the 580 nm setup only, a separate calibration fibre is available to acquire simultaneous ThAr calibration spectra. This allows very accurate radial velocity determinations. In this configuration, 7 fibres remain available for targets on sky.

Spectrograph	Mode	N. of Objects	Aperture (arcsecs)	Resolving power	Spectral Band [nm]
UVES	Red arm	8	1.0	47000	200
UVES	Red arm	7+1(calibration)	1.0	47000	200

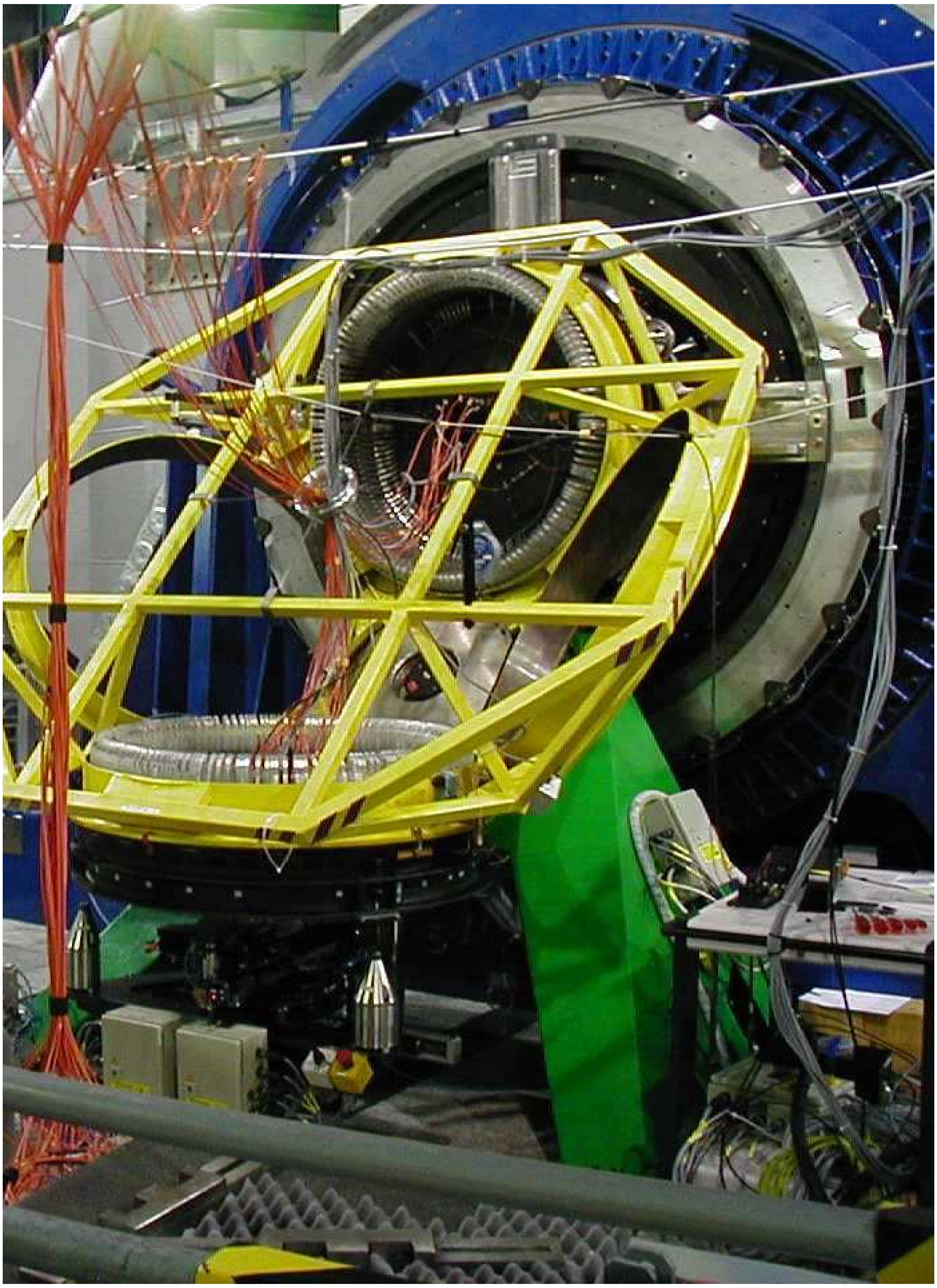

Figure 4.1.1: A photo of the OzPoz fibre positioner. While one plate is observing, the other one is positioning the fibres for the subsequent observations.

5 Quick start

This section describes the most immediate usage of the FLAMES-UVES pipeline recipes.

5.1 FLAMES-UVES pipeline recipes

The current FLAMES-UVES pipeline is based on a set of 7 stand-alone recipes (and uses 2 recipes from the UVES pipeline to combine biases and darks) involved in the data reduction cascade:

uves_cal_mbias creates a master bias frame.

uves_cal_mdark creates a master dark frame.

flames_cal_predict implements the UVES physical model (and apply a shift to reflect the offset between the calibration fibre and the order centre).

flames_cal_orderpos defines the calibration fibre order positions.

flames_cal_mkmaster creates master slit flat field frames.

flames_cal_prep_sff_ofpos process master slit flats and odd-even-all fibre flats to determine the order table tracing all fibres and additional frames needed to extract the science data.

flames_cal_wavecal performs the wavelength calibration.

flames_obs_scired reduces a science frame.

Alternatively, to run the full data reduction chain, a user not willing to check results step-by-step may use the **flames_obs_redchain** recipe.

5.2 An introduction to Gasgano and EsoRex

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

The process of data classification and association is known as data 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 *Gasgano* tool.

Gasgano is a data management tool that simplifies the data organisation process, offering automatic data classification and making the data association easier (*even if 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.

5.2.1 Using Gasgano

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

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

```
gasgano $HOME/gasgano/config/UVES.prefs &
```

where we have passed as first optional argument explicitly the UVES preference file which defines proper defaults for UVES data reduction. The user may like to realias the gasgano command to the previous command. The *Gasgano* main window will appear. On Figure 5.2.1 (next page), a view on a set of UVES data is shown as an example. *Gasgano* can be pointed to the directories where the data to be handled are located using the navigation panels accessible via the *Add/Remove Files* entry of the *File* menu (shown on the upper left of the figure).

The data are hierarchically organised as preferred by the user. After each file name are shown the classification and the values of the following FITS keywords (we omit the prefix HIERARCH.ESO):

Keyword name	Purpose
CLASSIFICATION	Data classification
OBS.TARG.NAME	Observation Block target name
EXPTIME	Exposure time
DATE	Observing date
DET.CHIPS	# of chips in detector array
INS.MODE	Instrument mode used
INS.GRAT1.NAME	Instrument grating name (for blue arm)
INS.GRAT1.WLEN	Instrument setting central wavelength (for blue arm)
INS.SLIT2.WID	Instrument slit width (for blue arm)
INS.GRAT2.NAME	Instrument grating name (for red arm)
INS.GRAT2.WLEN	Instrument setting central wavelength (for red arm)
INS.SLIT3.WID	Instrument slit width (for red arm)
DET.READ.SPEED	Readout speed
DET.WIN1.BINX	Binning factor along X
DET.WIN1.BINY	Binning factor along Y
INS.OBSPLATE	Observation plate
OCS.SIMCAL	Simultaneous calibration

The CLASSIFICATION field provides either the value of the PRO.CATG, for pipeline products; or a user defined file classification, if provided, defined in the classification rule file, which can be accessed by Gasgano from the Tools –> Classification rules... tab; or the default value "UNDEFINED". File classification rules are selection rules which assign to a FITS file a classification based on the value of a few FITS keywords, usually the DPR.TYPE, DPR.TECH, DPR.CATG values, which respectively define the file data type, acquisition technique and category, and from the keyword values of INS.GRATi.WLEN, DET.CHIPS, INS.SLITi.NAME. Additional relevant keywords are DET.WIN1.BINX/Y, INS.MODE, INS.GRATi.NAME, INS.OBSPLATE, OCS.SIMCAL. Those relevant keywords are indicated by Gasgano either in the file section, or by selecting each file, in the section which shows the FITS file header content. Alternatively the user can access those FITS keyword values from the command line with the command

dfits file.fits | grep FITS.KEY.NAME

More information about a single frame can be obtained by clicking on its name: the corresponding FITS file header will be displayed on the bottom panel, where specific keywords can be opportunely filtered and searched. Images and tables may be easily displayed using the viewers specified in the appropriate *Preferences* fields. Such a field allows also to set the file filter, which should point to the $HOME/gasgano/config/UVES.rul. This rule file provides simple filtering rules to select UVES data corresponding to a given standard data reduction setting.

Frames can be selected from the main window for being processed by the appropriate recipe: on Figure 5.2.2, the standard star frame, previously produced master bias and master flat frames, together with a line and an order tables, a table with the reference standard star spectra and one with the atmospheric dispersion are all selected and sent to the *uves_cal_response* recipe. This will open a *Gasgano* recipe execution window (see Figure 5.2.3), having all the specified files listed in its *Input Frames* panel.

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

At this point the recipe can be launched by pressing the *Execute* button. Messages from the running recipe will appear on the *Log Messages* panel at bottom, and in case of successful completion the products will be listed on the *Output Frames* panel, where they can be easily viewed and located back on the Gasgano main window. To produce useful plots the user need to set to 'gnuplot -persist' the **plotter** recipe parameter value (and have a valid installation of gnuplot package, and the gnuplot command available in the PATH). Please refer to the *Gasgano User's Manual* [7] for a more complete description of the *Gasgano* interface.

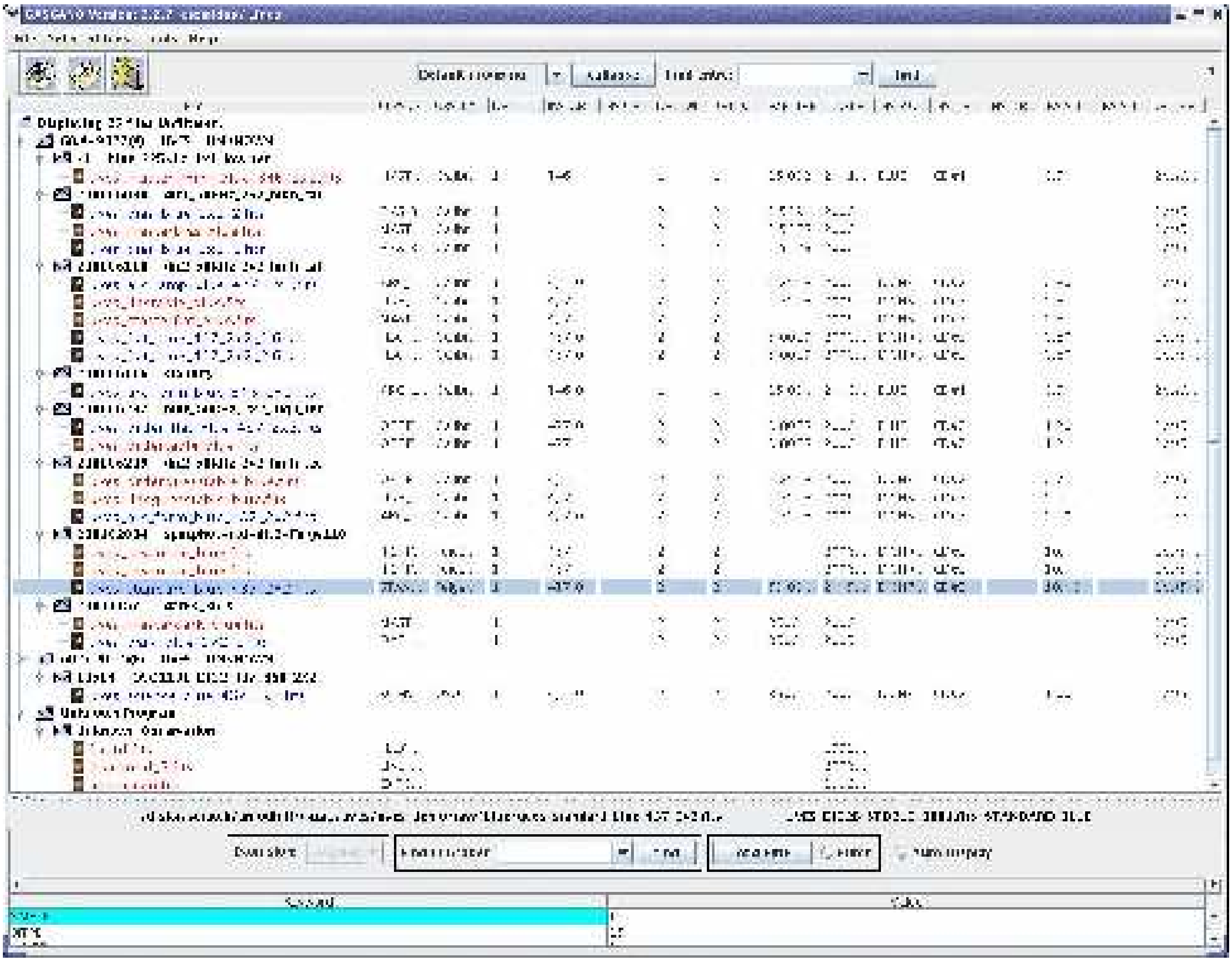

Figure 5.2.1: *The Gasgano main window.*

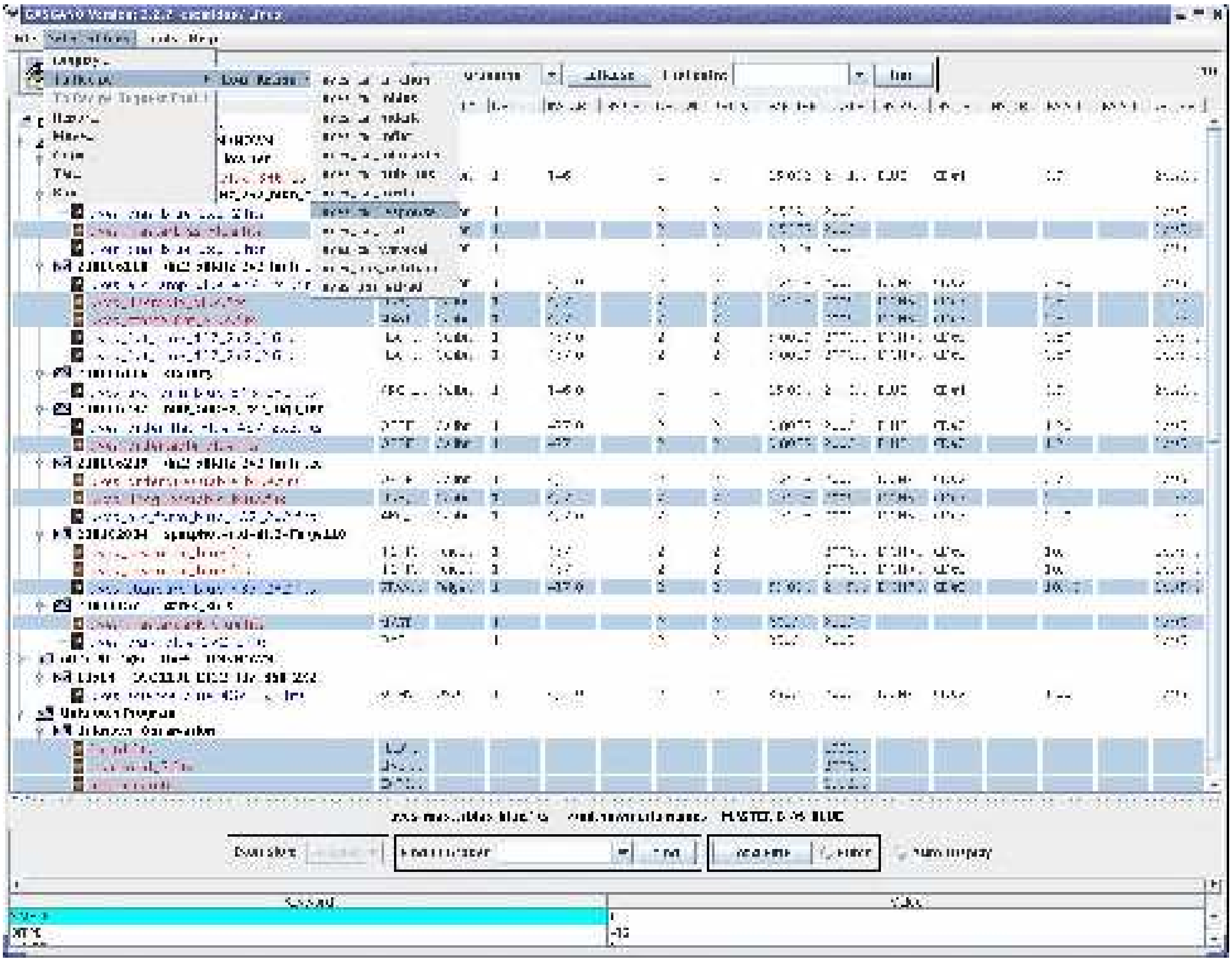

Figure 5.2.2: *Selecting files to be processed by a UVES pipeline recipe.*

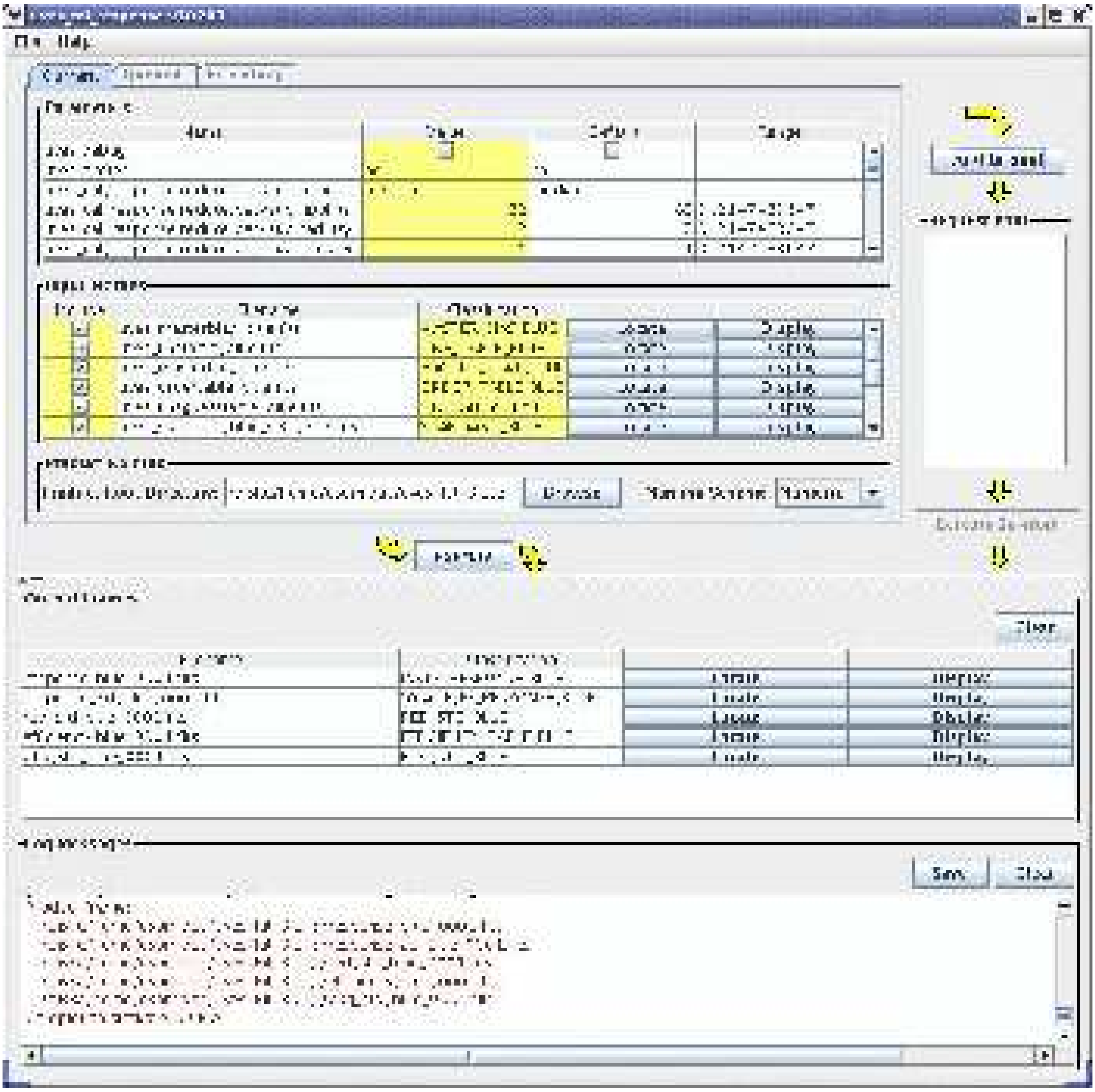

Figure 5.2.3: *The Gasgano recipe execution window.*

5.2.2 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It can be used in data reduction scripts for the automation of processing tasks. Unlike when using *Gasgano*, the user must classify and associate the data using the information contained in the FITS header keywords (see Section 7, page 35). 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 [1] in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [2] DO categories for the supported FLAMES-UVES input raw frames are indicated in section 7.8.

```
/path_raw/UVES.2004-08-14T10:20:56.497.fits  FIB_ARC_LAMP_RED
/path_pro/ordertable_redl.fits                FIB_ORDEF_TABLE_REDL
/path_pro/lineguess_redl.fits                 FIB_LIN_GUE_REDL
/path_pro/ordertable_redu.fits                FIB_ORDEF_TABLE_REDU
/path_pro/lineguess_redu.fits                 FIB_LIN_GUE_REDU
/path_pro/thargood_3.fits                     LINE_REFER_TABLE
```

It contains for each input frame the full path file name and its DO category. The pipeline recipe will access the listed files when required by the reduction algorithm.

Note that the FLAMES-UVES pipeline recipes do not verify in any way the correctness of the classification tags specified by the user in the SOF. In the above example, the recipe *flames_cal_wavecal* will treat the frame `/path_raw/UVES.2004-08-14T10:20:56.497.fits` as an FIB_ARC_LAMP_RED, the frame `/path_pro/ordertable_redl.fits` as a FIB_ORDEF_TABLE_REDL, etc., even when they do not contain this type of data. The recipe will also assume that all frames are associated correctly, *i.e.*, that they all come from the same arm, dichroic and bin setting.

The reason of this lack of control is that the FLAMES-UVES recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 5.2.1, page 17). Also this lack of control allows the user to reduce e.g. an arc lamp frame pretending it is a science frame.

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

EsoRex syntax: The basic syntax to use EsoRex is the following:

esorex [esorex_options] recipe_name [recipe_options] set_of_frames

To get more information on how to customise EsoRex (see also [6]) run the command:

[1] The indicated *DO category* is a label assigned to any data type after it has been classified, which is then used to identify the frames listed in the *set-of-frames*

[2] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 5.2.3, page 21).

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

EsoRex searches for recipes in the directory specified by the option

esorex - -recipe-dir=installation_directory

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

esorex - -params recipe_name

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

esorex - -help recipe_name

To get more details about the given recipe give the command at the shell prompt:

esorex - -man-page recipe_name

Recipe configuration: To each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe.[3] The configuration files are normally generated in the directory `$HOME/.esorex`, and have the same name as the recipe to which they are related, with the filename extension `.rc`. For instance, the recipe *flames_cal_wavecal* has its *EsoRex* generated configuration file named `flames_cal_wavecal.rc`, and is generated with the command:

esorex - -create-config flames_cal_wavecal

The definition of one parameter of a recipe may look like this:

```
# --tolerance
# Tolerance of fit. If positive, 'tolerance' is in pixel units. If negative,
# abs('tolerance') is in wavelength units. Lines with residuals worse than
# the tolerance are excluded from the final fit. Unlike in previous versions,
# this parameter is not corrected for CCD binning.
flames_cal_wavecal.calibrate.tolerance=0.6
```

In this example, the parameter `flames_cal_wavecal.calibrate.tolerance` is set to the value `0.6`. In the configuration file generated by *EsoRex*, one or more comment lines are added containing information about the possible values of the parameter, and an alias that could be used as a command line option.

The command

esorex - -create-config recipe_name

generates a default configuration file **recipe_name.rc** in the directory **$HOME/.esorex**[4].

A recipe configuration file different from the default one can be specified on the command line:

esorex - -recipe-config=my_alternative_recipe_config

[3] The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 5.2.3, page 21).

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

Recipe parameters are provided in section 10 and their role is described in Section 11.

More than one configuration file may be maintained for the same recipe but, in order to be used, a configuration file not located under `$HOME/.esorex`, or having a name different from the recipe name, should be explicitly specified when launching a recipe.

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

esorex flames_cal_wavecal flames_cal_wavecal.sof

The recipe parameters can be modified either by editing directly the used configuration file, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name and before the SOF name, and they will supersede the system defaults and/or the configuration file settings. For instance, to set the *flames_cal_wavecal* recipe *tolerance* parameter to `0.07`, the following should be typed:

esorex flames_cal_wavecal --tolerance=0.07 flames_cal_wavecal.sof

Every recipe provides a parameter *debug* which – when enabled – causes intermediate results to be saved to the local directory. This allows more detailed inspection of the recipe processing

esorex flames_cal_wavecal --debug flames_cal_wavecal.sof

Basic plotting functionality can be enabled on systems which have the gnuplot tool:

esorex flames_cal_wavecal --plotter='gnuplot -persist' flamses_cal_wavecal.sof

For more advanced visualisation a dedicated FITS viewer should be used.

For more information on *EsoRex*, see `www.eso.org/cpl/esorex.html`.

5.3 Example of data reduction using EsoRex

Here we provide an example of data reduction for data obtained with the FLAMES-UVES.

The simplest and least interactive way to reduce the data is to create a SOF (set-of-frames) file from all raw calibrations, master calibrations and raw science frames:

```
/path_raw/uves_bias_red1.fits    BIAS_RED
/path_raw/uves_bias_red2.fits    BIAS_RED
/path_raw/uves_bias_red3.fits    BIAS_RED
/path_raw/uves_bias_red4.fits    BIAS_RED
/path_raw/uves_bias_red5.fits    BIAS_RED

/path_raw/uves_dark_red1.fits    DARK_RED
/path_raw/uves_dark_red2.fits    DARK_RED
/path_raw/uves_dark_red3.fits    DARK_RED

/path_raw/uves_flat_set1_red1.fits   SFLAT_RED
```

```
/path_raw/uves_flat_set1_red2.fits   SFLAT_RED
/path_raw/uves_flat_set1_red3.fits   SFLAT_RED
/path_raw/uves_flat_set2_red1.fits   SFLAT_RED
/path_raw/uves_flat_set2_red2.fits   SFLAT_RED
/path_raw/uves_flat_set2_red3.fits   SFLAT_RED
/path_raw/uves_flat_set3_red1.fits   SFLAT_RED
/path_raw/uves_flat_set3_red2.fits   SFLAT_RED
/path_raw/uves_flat_set3_red3.fits   SFLAT_RED

/path_raw/flames_uves_odd_red.fits  FIB_FF_ODD_RED
/path_raw/flames_uves_even_red.fits FIB_FF_EVEN_RED
/path_raw/flames_uves_all_red.fits  FIB_FF_ALL_RED

/path_raw/flames_uves_arc_form_red.fits FIB_ARC_FORM_RED

/path_raw/flames_uves_order_flat_red.fits    FIB_ORDER_FLAT_RED

/path_raw/flames_uves_arc_lamp_red.fits FIB_ARC_LAMP_RED

/path_raw/flames_uves_science_red.fits  FIB_SCI_RED
/path_raw/flames_uves_science_red.fits  FIB_SCI_SIM_RED
/path_raw/flames_uves_science_red.fits  FIB_SCI_COM_RED

/path_ref/thargood_3.fits         LINE_REFER_TABLE
/path_ref/flxstd.fits             FLUX_STD_TABLE
/path_ref/atmoexan.fits           EXTCOEFF_TABLE
```

Then run

esorex flames_obs_redchain flames_obs_redchain.sof

which will execute the necessary recipes and create the pipeline products listed in following section.

In this example five raw bias frames are provided; therefore the uves_cal_mbias recipe will be executed in order to produce the master bias frame. For the same reason the uves_cal_mdark and flames_cal_mflat will be executed. If the user already has a good master frame (master bias, master dark or master slit flat), may be convenient to use it in place of the corresponding raw frames, and therefore the corresponding master creation recipe (uves_cal_mbias, uves_cal_mdark, flames_cal_mflat) will not be executed.

In the following a typical step-by-step data reduction procedure is described. [5] Figure 5.3.1 gives an overview.

We suggest the user to group the data according to detector arm, dichroic and detector binning setting. `/path_ref` indicates the full path to the source tree directory containing reference ancillary data, `/path_pro` indicates the full path to the source tree directory containing product data.

Formatcheck: these frames are characterized by DPR.TYPE='LAMP,FMTCHK',

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

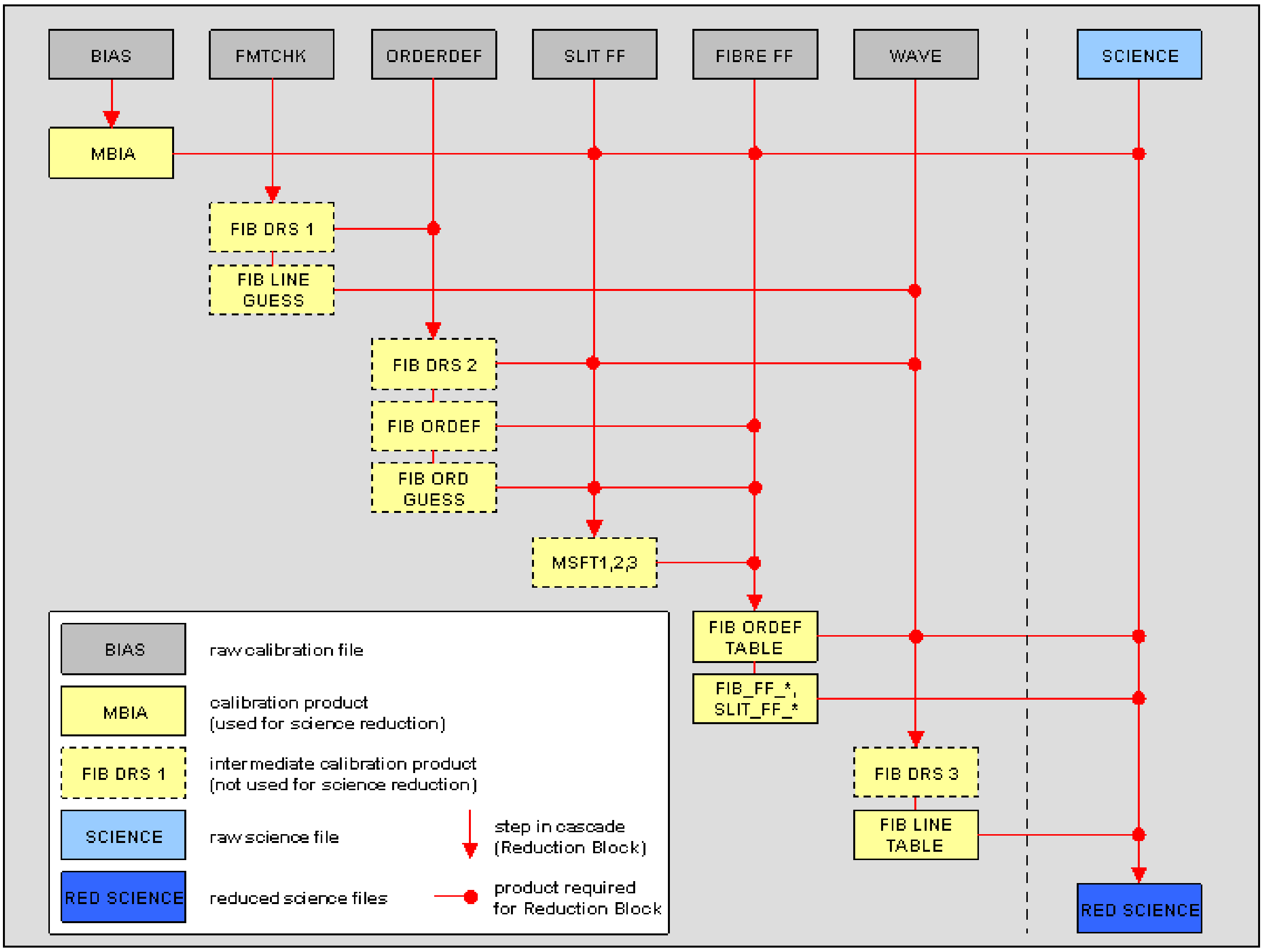

Figure 5.3.1: *The FLAMES-UVES calibration cascade.*

```
/path_raw/flames_uves_arc_lamp_form_blue.fits FIB_ARC_LAMP_FORM_RED
```

Single fibre order tracing flat field frames: these frames are characterized by DPR.TYPE='LAMP,ORDERDEF,SimCal'

```
/path_raw/flames_uves_order_flat_blue.fits FIB_ORDER_FLAT_RED
/path_raw/ordertable_redl.fits            FIB_ORD_GUE_REDL
/path_raw/order_def_redl.fits             FIB_ORDEF_REDL
/path_raw/ordertable_redu.fits            FIB_ORD_GUE_REDU
/path_raw/order_def_redu.fits             FIB_ORDEF_REDU
```

Arc lamp frames to compute the wavelength calibration: these frames have DPR.TYPE respectively equal to 'LAMP,WAVE,OzPoz'.

```
/path_raw/flames_uves_arc_lamp_red.fits FIB_ARC_LAMP_RED
```

Bias frames: these frames are characterized by DPR.TYPE='BIAS'

```
/path_raw/uves_bias_red1.fits  BIAS_RED
/path_raw/uves_bias_red2.fits  BIAS_RED
/path_raw/uves_bias_red3.fits  BIAS_RED
/path_raw/uves_bias_red4.fits  BIAS_RED
/path_raw/uves_bias_red5.fits  BIAS_RED
```

Flat field lamp frames: these frames are characterized by DPR.TYPE='LAMP,SFLAT'

```
/path_raw/uves_flat_set1_red1.fits SFLAT_RED
/path_raw/uves_flat_set1_red2.fits SFLAT_RED
/path_raw/uves_flat_set1_red3.fits SFLAT_RED
/path_raw/uves_flat_set2_red1.fits SFLAT_RED
/path_raw/uves_flat_set2_red2.fits SFLAT_RED
/path_raw/uves_flat_set2_red3.fits SFLAT_RED
/path_raw/uves_flat_set3_red1.fits SFLAT_RED
/path_raw/uves_flat_set3_red2.fits SFLAT_RED
/path_raw/uves_flat_set3_red3.fits SFLAT_RED
```

science frames: these frames are characterized by DPR.TYPE='OBJECT'.

```
/path_raw/flames_uves_science_red.fits  FIB_SCI_RED
```

In the following examples we assume that pipeline product filenames are the original as set by the pipeline. This corresponds to have the parameter **esorex.caller.suppress-prefix** in the EsoRex configuration file ($HOME/.esorex/esorex.rc) set to TRUE. Otherwise EsoRex will rename the pipeline producs using a common prefix (set by the parameter **esorex.caller.output-prefix**), a four digits increasing number, and terminating the FITS file with the extention ".fits". We suggest to verify to have the flag *readonly* set to FALSE, if the user would like to run the same recipe several times with EsoRex having standard values for product files. This setting allows the pipeline to overwrite previously generated products [6].

1. Generate guess order and line tables. Formatcheck frames are listed together with the needed calibration frames in an ASCII file, flames_cal_predict.sof. This file will look like as follows:

   ```
   /path_raw/flames_uves_arc_lamp_form_red.fits      FIB_ARC_LAMP_FORM_RED
   /path_ref/thargood_3.fits                         LINE_REFER_TABLE
   ```

   Then the user can generate the guess order and line tables with the command

   esorex flames_cal_predict flames_cal_predict.sof

   This command will generate two files (in the following table FITS files have extention .fits):

[6] By default installation in the EsoRex configuration file ($HOME/.esorex/esorex.rc) the flag *suppress-prefix* is set to FALSE and the flag *readonly* is set to FALSE, a possible combination, in which case pipeline product filenames will have a prefix out_, an increasing four digit number, and extention .fits.

default recipe filename	format	PRO.CATG	short description
lineguesstable_redl	table	FIB_LIN_GUE_REDL	guess line table
orderguesstable_redl	table	FIB_ORD_GUE_REDL	guess order table
lineguesstable_redu	table	FIB_LIN_GUE_REDU	guess line table
orderguesstable_redu	table	FIB_ORD_GUE_REDU	guess order table

and generates the ASCII files lineguesstable_redl-0.paf, and lineguesstable_redu-0.paf which logs QC parameters.

mv *.fits *.paf /path_pro

PAF files are ASCII files containing quality control information.

2. Generate an order table. A set of narrow slit raw flat field frames may be put in the ASCII file flames_cal_orderpos.sof.

```
/path_raw/flames_uves_order_flat_red.fits FIB_ORDEF_RED
```

The user can generate order tables and reoriented single fibre order flats with the command:

esorex flames_cal_orderpos flames_cal_orderpos.sof

This command will generate the following products:

default recipe filename	format	PRO.CATG	short description
ordertable_redl	table	FIB_ORD_GUE_REDL	simultaneous calibration fibre order table
order_def_redl	table	FIB_ORDEF_REDL	simultaneous calibration fibre order frame
ordertable_redu	table	FIB_ORD_GUE_REDU	simultaneous calibration fibre order table
order_def_redu	table	FIB_ORDEF_REDU	simultaneous calibration fibre order frame

and generates the ASCII files ordertable_redl-0.paf, ordertable_redu-0.paf which logs QC parameters.

mv *.fits *.paf /path_pro

3. Then one selects the raw biases and lists them in an ASCII file uves_cal_mbias.sof:

```
/path_raw/uves_bias_red1.fits  BIAS_RED
/path_raw/uves_bias_red2.fits  BIAS_RED
/path_raw/uves_bias_red3.fits  BIAS_RED
/path_raw/uves_bias_red4.fits  BIAS_RED
/path_raw/uves_bias_red5.fits  BIAS_RED
```

The command:

esorex uves_cal_mbias uves_cal_mbias.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description

masterbias_redl	2d image (pix-pix)	MASTER_BIAS_REDL	master bias
masterbias_redu	2d image (pix-pix)	MASTER_BIAS_REDU	master bias

and generates the ASCII file masterbias_redl-0.paf, masterbias_redu-0.paf which logs QC parameters.

mv *.fits *.paf /path_pro

4. Then one selects the raw darks and list them in an ASCII file uves_cal_mdark.sof.

```
/path_raw/uves_dark_red1.fits  DARK_RED
/path_raw/uves_dark_red2.fits  DARK_RED
/path_raw/uves_dark_red3.fits  DARK_RED
/path_pro/masterbias_redl.fits MASTER_BIAS_REDL
/path_pro/masterbias_redu.fits MASTER_BIAS_REDU
```

The command:

esorex uves_cal_mdark uves_cal_mdark.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
masterdark_redl	2d image (pix-pix)	MASTER_DARK_REDL	master dark
masterdark_redu	2d image (pix-pix)	MASTER_DARK_REDU	master dark

and generates the ASCII file masterdark_redl-0.paf, masterdark_redu-0.paf which logs QC parameters.

mv *.fits *.paf /path_pro

5. Then one selects the raw slit flat fields and list them in an ASCII file flames_cal_mflat.sof together with some master calibrations and previously obtained products:

```
/path_raw/flames_uves_flat_set1_red1.fits SFLAT_RED
/path_raw/flames_uves_flat_set1_red2.fits SFLAT_RED
/path_raw/flames_uves_flat_set1_red3.fits SFLAT_RED
/path_raw/flames_uves_flat_set2_red1.fits SFLAT_RED
/path_raw/flames_uves_flat_set2_red2.fits SFLAT_RED
/path_raw/flames_uves_flat_set2_red3.fits SFLAT_RED
/path_raw/flames_uves_flat_set3_red1.fits SFLAT_RED
/path_raw/flames_uves_flat_set3_red2.fits SFLAT_RED
/path_raw/flames_uves_flat_set3_red3.fits SFLAT_RED
/path_pro/masterbias_redl.fits            MASTER_BIAS_REDL
/path_pro/ordertable_redl.fits            ORDER_TABLE_REDL
/path_pro/masterbias_redu.fits            MASTER_BIAS_REDU
/path_pro/ordertable_redu.fits            ORDER_TABLE_REDU
```

The command:

esorex flames_cal_mflat flames_cal_mflat.sof

will generate the following products:

default recipe filename	format	PRO.CATG	Note
masterflat_set1_redl	2d image (pix-pix)	MASTER_SFLAT_REDL	master flat
masterflat_set2_redl	2d image (pix-pix)	MASTER_SFLAT_REDL	master flat
masterflat_set3_redl	2d image (pix-pix)	MASTER_SFLAT_REDL	master flat
masterflat_bkg_redl	2d image (pix-pix)	BKG_FLAT_REDL	inter-order background
masterflat_set1_redu	2d image (pix-pix)	MASTER_SFLAT_REDU	master flat
masterflat_set2_redu	2d image (pix-pix)	MASTER_SFLAT_REDU	master flat
masterflat_set3_redu	2d image (pix-pix)	MASTER_SFLAT_REDU	master flat
masterflat_bkg_redu	2d image (pix-pix)	BKG_FLAT_REDU	inter-order background

and generates the ASCII file masterflat_redl-0.paf, masterflat_redu-0.paf which log QC parameters.

mv *.fits *.paf /path_pro

6. Then the order table for all fibre traces is determined. A set of three raw fibre frames illuminating respectively the odd, even, all fibre frames are put in an the ASCII file flames_cal_prep_sff_ofpos.sof.

```
/path_raw/flames_uves_odd_red.fits   FIB_FF_ODD_RED
/path_raw/flames_uves_even_red.fits  FIB_FF_EVEN_RED
/path_raw/flames_uves_all_red.fits   FIB_FF_ALL_RED
/path_pro/masterbias_redl.fits       MASTER_BIAS_REDL
/path_pro/ordertable_redl.fits       FIB_ORD_GUE_REDL
/path_pro/masterflat_set1_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterflat_set2_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterflat_set2_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterbias_redu.fits       MASTER_BIAS_REDU
/path_pro/ordertable_redu.fits       FIB_ORD_GUE_REDU
/path_pro/masterflat_set1_redu.fits  MASTER_SFLAT_REDU
/path_pro/masterflat_set2_redu.fits  MASTER_SFLAT_REDU
/path_pro/masterflat_set2_redu.fits  MASTER_SFLAT_REDU
```

Note that the input master bias frames is optional recommended input.

The command

esorex flames_cal_prep_sff_ofpos flames_cal_prep_sff_ofpos.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
xt_odd_l	table	FIB_FF_ODD_INFO_TAB	info table
xt_even_l	table	FIB_FF_EVEN_INFO_TAB	info table
xt_all_l	table	FIB_FF_ODD_INFO_TAB	info table
slitff_common_redl	table	SLIT_FF_COM_REDL	slitff common frame
slitff_norm_redl	table	SLIT_FF_NOR_REDL	slitff common frame

slitff_dtc_redl	table	SLIT_FF_DTC_REDL	slitff data frame
slitff_bpc_redl	table	SLIT_FF_BPC_REDL	slitff badpixel frame
slitff_sgc_redl	table	SLIT_FF_SGC_REDL	slitff sigma frame
fibff_common_redl	table	FIB_FF_COM_REDL	fibff common frame
fibff_norm_redl	table	FIB_FF_NOR_REDL	fibff common frame
fibff_dtc_redl	table	FIB_FF_DTC_REDL	fibff data frame
fibff_bpc_redl	table	FIB_FF_BPC_REDL	fibff badpixel frame
fibff_sgc_redl	table	FIB_FF_SGC_REDL	fibff sigma frame
fibff_bnc_redl	table	FIB_FF_BNC_REDL	fibff bound frame
fib_ordef_redl	table	FIB_ORDEF_TABLE_REDL	fibre order table
slitff_common_redu	table	SLIT_FF_COM_REDU	slitff common frame
slitff_norm_redu	table	SLIT_FF_NOR_REDU	slitff common frame
slitff_dtc_redu	table	SLIT_FF_DTC_REDU	slitff data frame
slitff_bpc_redu	table	SLIT_FF_BPC_REDU	slitff badpixel frame
slitff_sgc_redu	table	SLIT_FF_SGC_REDU	slitff sigma frame
fibff_common_redu	table	FIB_FF_COM_REDU	fibff common frame
fibff_norm_redu	table	FIB_FF_NOR_REDU	fibff common frame
fibff_dtc_redu	table	FIB_FF_DTC_REDU	fibff data frame
fibff_bpc_redu	table	FIB_FF_BPC_REDU	fibff badpixel frame
fibff_sgc_redu	table	FIB_FF_SGC_REDU	fibff sigma frame
fibff_bnc_redu	table	FIB_FF_BNC_REDU	fibff bound frame
fib_ordef_redu	table	FIB_ORDEF_TABLE_REDU	fibre order table

and generates the ASCII file masterflat_redl-0.paf, masterflat_redu-0.paf which log QC parameters.

mv *.fits *.paf /path_pro

7. Then the wavelength calibration is performed. A set of raw frames illuminated through a long slit by an arc lamp are put in the ASCII file flames_cal_wavecal.sof.

```
/path_raw/flames_uves_arc_lamp_red.fits   FIB_ARC_LAMP_RED
/path_pro/ordertable_redl.fits       FIB_ORDEF_TABLE_REDL
/path_pro/lineguesstable_redl.fits  LIN_GUE_REDL
/path_pro/masterbias_redl.fits       MASTER_BIAS_REDL
/path_pro/masterflat_redl.fits       MASTER_FLAT_REDL
/path_pro/ordertable_redu.fits       FIB_ORDEF_TABLE_REDU
/path_pro/lineguesstable_redu.fits  LIN_GUE_REDU
/path_pro/masterbias_redu.fits       MASTER_BIAS_REDU
/path_pro/masterflat_redu.fits       MASTER_FLAT_REDU
/path_ref/thargood_3.fits            LINE_REFER_TABLE
```

Note that the input master bias and master flat frames are optional recommended inputs.

The command

esorex flames_cal_wavecal flames_cal_wavecal.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
linetable_redl	table	FIB_LINE_TABLE_REDL	line table
linetable_redu	table	FIB_LINE_TABLE_REDU	line table

This table contains the solutions for each extracted window (sky-obj-sky) in several extensions according to the following schema:

```
Line table for trace 0, window #1 saved to extensions 1-3 of 'linetable_redl/redu'
Line table for trace 0, window #2 saved to extensions 4-6 of 'linetable_redl/redu'
Line table for trace 0, window #3 saved to extensions 7-9 of 'linetable_redl/redu'
```

and generates the ASCII file linetable_redl-0.paf, linetable_redu-0.paf which logs QC parameters.

mv *.fits *.paf /path_pro

8. Finally, the raw science frame is reduced.The raw science frame is listed together with master calibration products in the following ASCII file flames_obs_scired.sof:

```
/path_raw/flames_uves_science_red.fits FIB_SCI_RED
/path_pro/slitff_com_redl.fits         SLIT_FF_COM_REDL
/path_pro/slitff_nor_redl.fits         SLIT_FF_NOR_REDL
/path_pro/slitff_dtc_redl.fits         SLIT_FF_DTC_REDL
/path_pro/slitff_bpc_redl.fits         SLIT_FF_BPC_REDL
/path_pro/slitff_sgc_redl.fits         SLIT_FF_SGC_REDL
/path_pro/fibff_com_redl.fits          FIB_FF_COM_REDL
/path_pro/fibff_nor_redl.fits          FIB_FF_NOR_REDL
/path_pro/fibff_dtc_redl.fits          FIB_FF_DTC_REDL
/path_pro/fibff_bpc_redl.fits          FIB_FF_BPC_REDL
/path_pro/fibff_sgc_redl.fits          FIB_FF_SGC_REDL
/path_pro/fibff_bnc_redl.fits          FIB_FF_BNC_REDL
/path_pro/fib_ordef_redl.fits          FIB_ORDEF_TABLE_REDL
/path_pro/slitff_com_redu.fits         SLIT_FF_COM_REDU
/path_pro/slitff_nor_redu.fits         SLIT_FF_NOR_REDU
/path_pro/slitff_dtc_redu.fits         SLIT_FF_DTC_REDU
/path_pro/slitff_bpc_redu.fits         SLIT_FF_BPC_REDU
/path_pro/slitff_sgc_redu.fits         SLIT_FF_SGC_REDU
/path_pro/fibff_com_redu.fits          FIB_FF_COM_REDU
/path_pro/fibff_nor_redu.fits          FIB_FF_NOR_REDU
/path_pro/fibff_dtc_redu.fits          FIB_FF_DTC_REDU
/path_pro/fibff_bpc_redu.fits          FIB_FF_BPC_REDU
/path_pro/fibff_sgc_redu.fits          FIB_FF_SGC_REDU
/path_pro/fibff_bnc_redu.fits          FIB_FF_BNC_REDU
/path_pro/fib_ordef_redu.fits          FIB_ORDEF_TABLE_REDU
```

The command:

esorex flames_obs_scired flames_obs_scired.sof
will generate the following products:

default recipe file name	format	PRO.CATG	short description
fxb_l_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDL	extracted, flatfielded raw frame
wfxb_l_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDL	rebinned, extracted, flatfielded raw frame
fxb_l_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDL	error frame
wfxb_l_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDL	error frame
mwfxb_l_rawsig000i	1d (wav) image	MWXB_SCI_RAW_REDL	merged, rebinned, extracted, flat fielded raw frame
fxb_l_000i	2d (pix-ord) image	XB_SCI_REDL	extracted, flatfielded raw frame
wfxb_l_000i	2d (wav-ord) image	WXB_SCI_REDL	rebinned, extracted, flatfielded raw frame
fxb_l_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDL	error frame
wfxb_l_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDL	error frame
mwfxb_l_sig000i	1d (wav) image	MWXB_SCI_REDL	merged, rebinned, extracted, flat fielded raw frame
fxb_u_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDU	extracted, flatfielded raw frame
wfxb_u_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDU	rebinned, extracted, flatfielded raw frame
fxb_u_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDU	error frame
wfxb_u_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDU	error frame
mwfxb_u_rawsig000i	1d (wav) image	MWXB_SCI_RAW_REDU	merged, rebinned, extracted, flat fielded raw frame
fxb_u_000i	2d (pix-ord) image	XB_SCI_REDU	extracted, flatfielded raw frame
wfxb_u_000i	2d (wav-ord) image	WXB_SCI_REDU	rebinned, extracted, flatfielded raw frame
fxb_u_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDU	error frame
wfxb_u_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDU	error frame
mwfxb_u_sig000i	1d (wav) image	MWXB_SCI_REDU	merged, rebinned, extracted, flat fielded raw frame

and generates the ASCII file resampled_ff_blue-0.paf which logs QC parameters.

6 Known problems

We suggest the user to execute the data reduction recipes using parameter defaults and all the reference and master calibrations indicated in this manual. The following is a list of currently-known problems with UVES recipes, and workarounds, if available:

- The radial velocity correlation is not computed.

7 Instrument Data Description

To reduce a science frame, the following calibration frames are needed:

- Fibre formatcheck
- Bias(es)
- Dark(s)
- Single Fibre Order definition
- Slit Flat field(s)
- Fibre flat fields having illuminated: odd, even, all fibres
- Fibre wavelength calibration

It is also necessary to have handy a reference (ThAr) line table, and in order to process standard stars, an atmospheric extinction table, and a standard star flux table.

7.1 Formatcheck frames

It is necessary to have a special formatcheck frame taken by illuminating the simultaneous calibration fibre with a ThAr lamp. This is used in combination with a physical model of UVES and the information contained in the FITS header and in a ThAr reference line table, to find a "guess" solution of the spectral format (order locations and wavelength calibration). This allows the user later on to obtain robust and automatic spectral format solutions.

7.2 Bias frames

Bias frames give the read out of the CCD detector of zero integration time with the shutter closed. Usually they are taken as a set of five exposures from which, through stacking, a Master Bias is created thus reducing the read out noise. This needs to be subtracted for example from the science frame to get the signal contribution from the source only.

7.3 Dark frames

Dark frames are measured occasionally, with the shutter closed. They are used to measure the dark current. They are measured for 1x1 binning with typical exposure times of 1h. There are also open-shutter DARKs (since December 2001). They include, in addition to the CCD dark current, contributions from the camera enclosure. Typical values are reported on the ESO Website under www.eso.org/observing/dfo/quality/UVES/qc/dark_qc1.html As the contribution of UVES dark exposures may be considered, in first approximation, negligible, they can be excluded from the data reduction chain as we will assume here in the following section.

7.4 Single fibre order definition frames

A ingle fibre order definition frame is a calibration exposure obtained with the calibration fibre illuminated by a continuum lamp. It is a very high signal-to-noise ratio fibre-echelle frame describing precisely the order location.

7.5 Fibre Flat Field (odd-enen-all) frames

The fibre flat-fields serve to locate the fibre positions, both relative to one another and absolute, within some tolerance; to know the cross-sectional profile of each fibre at each wavelength; and to determine the relative throughput of different fibres. These fibre flat-fields are only usable if the fibre traces fall in regions illuminated in at least one of the slit flat-fields; fibres whose light falls, even in part, in dark regions of the slit flat-fields are discarded. The data reduction software considers as really flat only a subregion of each slit flat field discarding a few pixels at the flat field border.

The fibres' images on the detector are so closely packed that the cross-sectional profiles of adjacent fibres overlap to some non-negligible extent. Therefore, a single flat-field frame, would not yield enough information on individual fibres to perform a correct data reduction. Instead, the DRS requires fibre flat-fields on which the fibres can be clearly separated, such as one containing only odd-numbered fibres and another containing only even-numbered fibres.

The order definition frame, if taken with the simultaneous fibre, could also be used as an additional flat-field frame covering the simultaneous calibration fibre. Such inclusion is necessary to be able to extract the simultaneous calibration fibre. In such a case one would add it in the pool of odd-even fibre flat fields (see below), and this set of frames would thus contain completely separable images of all fibres.

In addition to the above, an all-fibre flat-field is needed to correct for relative throughput differences between odd and even fibres. This can be replaced by a Nasmyth screen all-fibre flat-field for higher accuracy.

If it is planned to observe using the simultaneous calibration fibre, it is also necessary to have in the calibration data a ThAr frame with the ThAr lamp feeding all the same fibres used during the night (7OzPoz+1SimCal) as well as an all fibre flat-field with the flat-field lamp feeding all the same fibres used during the night (7OzPoz+1SimCal). This is the minimum fibre flat-field set usable by the DRS.

7.6 Slit Flat Field frames

Slit Flat Field frames are long slit exposures taken with a continuum lamp. They give information on the response of the detector, allowing to measure variations in efficiency at small (pixel-to-pixel), intermediate (fringing, in the far red) and large (the blaze function) scale. Usually they are taken as a three sets of three frames each at a given cross disperser setting, which after bias subtraction are stacked in a master to reject statistical outliers like cosmic ray events. The final Master Flat field is also background subtracted to eliminate diffused light from the orders in the inter-order regions. Science frames need to be corrected for pixel-to-pixel variations, interference fringes and the blaze function through division by the master flat field.

To properly evaluate the background level and avoid overlapping of orders, each slit flat field has to be taken with a limited slit aperture. To cover all the fibres (allowing also for some variability in their placement) several slit flat fields are thus necessary, taken with different offsets with respect to a central position and overlapping between each other.

7.7 Wavelength calibration frames

Wavelength calibration frames are fibre exposures taken (for UVES) illuminating all fibres with a ThAr arc lamp. They are used to find the wavelength calibration solution.

The wavelength calibration frames must be taken through the same set of fibres as the science frame.

We would like to emphasize the importance of having "proper" calibration data to be able to reduce the science observations with best quality. For example, if because of thermal shifts the available slit flat-fields do not cover the position where a (science-frame) fibre is located, this fibre will be discarded and its data is unusable. The same is true if fibre positions in the science-frame are shifted too much with respect to the corresponding positions of the fibre flat-field frames used to do the data reduction. This may occur in case of automatic on line pipeline data reduction on Paranal where usually the calibration data base is updated with a one month time scale. Moreover, the cross-sectional profiles of the fibres may change e. g. due to slightly different focusing. Since this, in turn, changes the contamination between adjacent fibres, a significantly different profile between calibrations and science data will lead to an incorrect deconvolution of the fibres. This release searches for possible instrument shifts along Y (i. e. perpendicular to dispersion) in the interval [-6,6] pixel. It satisfactorily reduced commissioning data which, due a still not completely stable instrument setup, were characterized by shifts between the science observation and the calibration in the range [-3.5,3.5]. It must be said that the source of such Y shift was discovered and fixed during an instrument software upgrade occurred before the beginning of FLAMES operations and so we do not expect the user will receive data affected by shifts between science and calibrations of more than one pixel. This release allows to interactively recover extraction also in cases of wider shifts (see help of recipe flames_obs_scired).

The FLAMES positioner uses two plates and two distinct sets of fibres, with two distinct entrance positions in UVES.

A simultaneous calibration fibre for precise radial velocity measurements is also available. Using this fibre the spectral format changes, including (at most) 7 OzPoz fibres and 1 SimCal fibre. Therefore, all calibration data need to be taken with the same FLAMES plate and ($\leq$ 8 OzPoz fibres)/($\leq$ 7 OzPoz fibres+1 SimCal fibre) configuration as the actual science frames to be reduced.

Although we introduced means to detect and appropriately treat saturated pixels in the data reduction, it might happen that a saturated frame affects the data reduction. More specifically, while isolated saturated pixels can be easily detected by the DRS and excluded from the subsequent data reduction, strongly saturated emission lines can produce a very irregular distribution of scattered light on the frame, causing a poor background subtraction and, as a consequence, wrong results in the determination of shifts to be applied to each fibre (correlation step). In particularly pathological cases, this may lead to a failure of the automatic data reduction procedure, and require manual intervention to properly extract the data.

7.8 Supported raw frames (keyword identifiers)

In this section we describe all possible types of raw frames for the different observing modes. More information on those data may be found on `www.eso.org/qc/uves`. The different frame types can be identified by the values of the DPR keywords of their FITS headers (see [5]). These keywords are generated by the FLAMES-UVES templates (for a description of the FLAMES-UVES templates see [9]). A given frame type can be processed by one or a few different dedicated pipeline recipes. The individual pipeline recipes are described in section 11. In most cases, reference data frames are needed to reduce a given frame. These reference data have to match the input frame in a number of instrument parameters (e.g. to apply a flat field correction to a science frame only a flat field frame taken in the same central wavelength, same slit length, etc. will be used for the correction). These parameters are listed under *relevant instrument parameters*.

The following raw frame types are possible:

- Relevant instrument parameters group 1 common to all raw frames:

Number of CCD chips:	`NCHIP`
Conversion e→ADU:	`ESO DET OUT1 CONAD`
x-binning:	`ESO DET WIN1 BINX`
y-binning:	`ESO DET WIN1 BINY`
Window start in x:	`ESO DET WIN1 STRX`
Window start in y:	`ESO DET WIN1 STRY`
No of pixels in x:	`ESO DET WIN1 NX`
No of pixels in y:	`ESO DET WIN1 NY`

- Relevant instrument parameters group 2 common to some raw frames:

Grating used:	`ESO INS GRATj ID`
Central wavelength used:	`ESO INS GRATj WLEN`
Slit width used:	`ESO INS SLITi WID`
Slit length used:	`ESO INS SLITi LEN`
Filter used:	`ESO INS FILTi ID`
Observation Plate used:	`INS OBSPLATE`
Simultaneous calibration:	`OCS SIMCAL`

`i`: 2 (`BLUE`), 3 (`RED`)
`j`: 1 (`BLUE`), 2 (`RED`)

Bias frames

- Template signature:
  `UVES_x_cal_bias`
  `(x: red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = BIAS`
- DO category:
  `BIAS_RED (NCHIP = 2)`

- Pipeline recipe: `uves_cal_mbias`
- Relevant instrument parameters: group 1.
- Reference: Figure 7.8.0 (a).

Dark frames

- Template signature:
  `UVES_x_cal_dark`
  `(x:  red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = DARK`
  `ESO DPR TECH = IMAGE`
- DO category:
  `DARK_RED (NCHIP = 2)`
- Pipeline recipe: `uves_cal_mdark`
- Relevant instrument parameters: group 1.

Order definition flat fields

- Template signature:
  `UVES_x_tec_orderdef`
  `(x:  red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,ORDERDEF,SimCal`
  `ESO DPR TECH = MOS`
- DO category:
  `ORDER_FLAT_RED (NCHIP = 2)`
- Pipeline recipe: `flames_cal_orderpos`
- Relevant instrument parameters groups 1, 2.
- Reference: Figure 7.8.0 (c).

Spectroscopic Slit flat fields

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: flatatt, flatfree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,SFLAT`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `FLAT_RED (NCHIP = 2)`
- Pipeline recipe: `flames_cal_mflat`
- Relevant instrument parameters groups 1, 2.
- Reference: Figure 7.8.0 (e).

Format check spectra

- Template signature:
  `UVES_x_tec_fmtchk`
  `(x: red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FMTCHK,SimCal`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_ARC_LAMP_FORM_RED (NCHIP = 2)`
- Pipeline recipe: `flames_cal_predict`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.8.0 (b).

Wavelength calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`

- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,WAVE,OzPoz`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_ARC_LAMP_RED` (`NCHIP = 2`)
- Pipeline recipe: `flames_cal_wavecal`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.8.0 (d).

Wavelength calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,WAVE,SimCal`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_ARC_LAMP_RED` (`NCHIP = 2`)
- Pipeline recipe: `flames_cal_wavecal`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.8.0 (d).

Odd Fibre flat calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FLAT,ODD`
  `ESO DPR TECH = MOS`

- DO category:
  `FIB_FF_ODD_RED` (`NCHIP = 2`)
- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.8.0 (d).

Even Fibre flat calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FLAT,EVEN`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_FF_EVEN_RED` (`NCHIP = 2`)
- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.8.0 (d).

All Fibre flat calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FLAT,ALL`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_FF_ALL_RED` (`NCHIP = 2`)
- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.8.0 (d).

Nasmyth screen Fibre flat calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FLAT,NASMYTH`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_FF_NASMYTH_RED` (`NCHIP = 2`)
- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.8.0 (d).

Science spectra

- Template signatures:
  `UVES_x_obs_y`
  `(x: blue, red, dic1, dic2)`
  `(y: exp, expfree)`
- DPR keywords:
  `ESO DPR CATG = SCIENCE`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_SCI_RED` (`ESO DPR TYPE = OBJECT,OzPoz`)
  `FIB_SCI_SIM_RED` (`ESO DPR TYPE = OBJECT,SimCal`)
  `FIB_SCI_COM_RED` (`ESO DPR TYPE = OBJECT,OzPoz`)
- Pipeline recipe: `uves_obs_scired`
- Relevant instrument parameters: groups 1, 2.

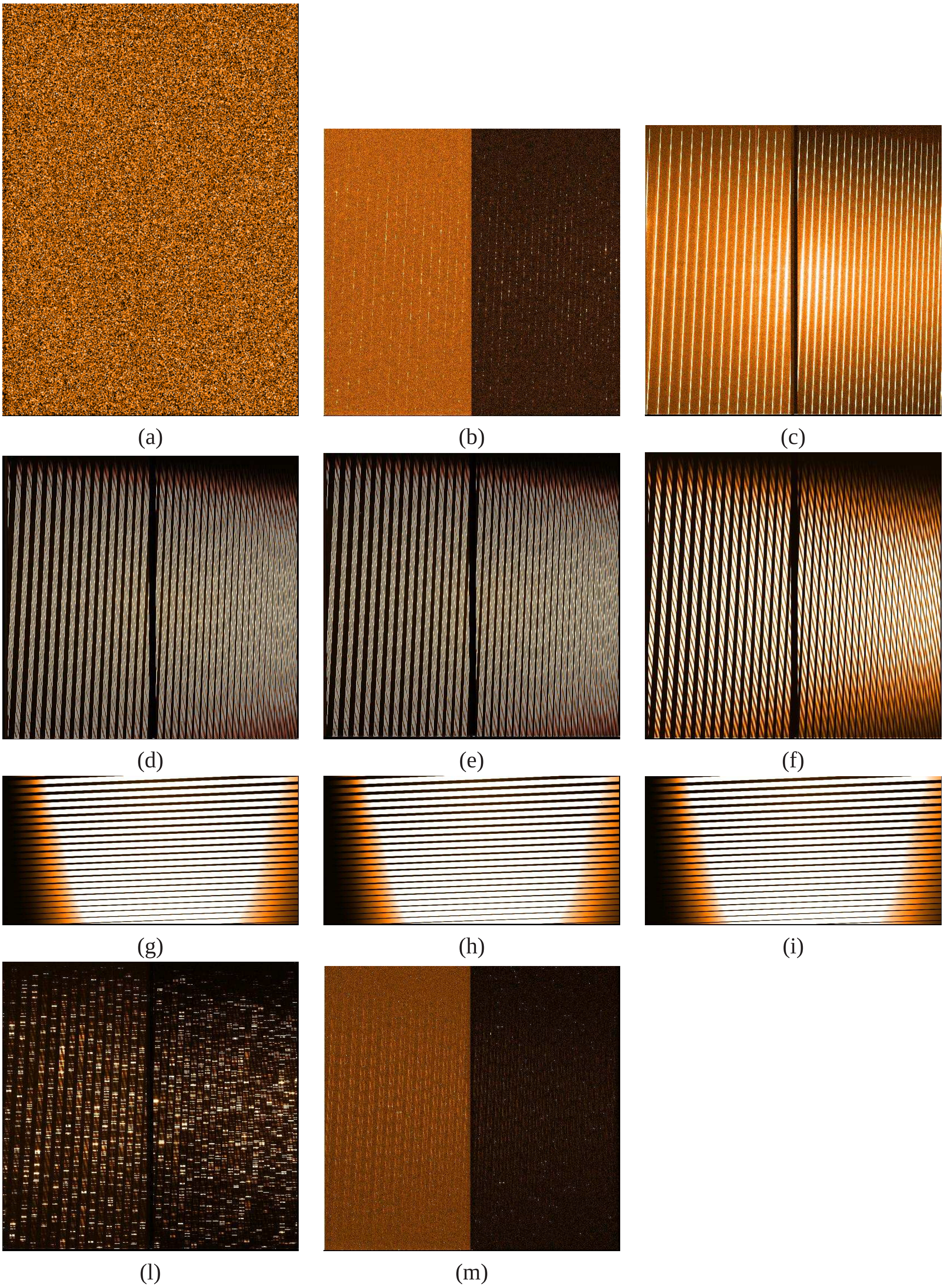

Figure 7.8.0: (a) a raw bias frame; (b) a raw formatcheck simcal fibre frame; (c) a raw order simcal fibre frame; (d) a raw odd fibres frame; (e) a raw even fibres frame; (f) a raw all fibres frame; (g) a master sflat frame (set1); (h) a master sflat frame (set2); (i) a master sflat frame (set3); (l) a raw arc lamp fibre frame; (m) a raw science frame.

8 Static Calibration Data

In the following section ancillary data required for UVES data reduction are listed. For each of them we indicate the corresponding value of the HIERARCH ESO PRO CATG, in short PRO.CATG, FITS keyword. This has to be used to identify the frames listed in the *Set of Frames* (see Section 5.2.2, page 22). More information on those data may be found on `www.eso.org/qc/uves`.

8.1 Line reference table

A reference list of arc lines is necessary to perform the wavelength calibration. Its PRO.CATG is LINE_REFER_TABLE. This frame is an input of the recipes uves_cal_predict and uves_cal_wavecal.

This release of the UVES pipeline provides a new reference line catalog (thargood_3.fits) more accurate than the previous one (thargood_2.fits). The kit contains both catalogs and the user is recommended to use the new one.

9 Data Reduction

In this section, after an overview of the main problems the data reduction needs to solve, we list the required data and the recipes which allow to solve them, giving the data reduction sequence necessary to reduce calibration and science data.

9.1 Data reduction overview

In order to fully reduce a set of FLAMES-UVES data, including calibration and science frames, the following data reduction problems need to be solved:

- The detector bias and dark current levels should be measured and subtracted.
- The echelle fiber order traces need to be determined in a robust manner.
- Be able to compute and correct for detector pixel to pixel gain variations and the blaze function.
- For each fibre perform the wavelength calibration in a robust and automatic manner.
- Reduce science fiber data recoving the cross order fibres contamination.

9.2 Required input data

To be able to reduce science data one needs to use raw, product data and recipes in a given sequence which provides all the necessary input to each pipeline recipe. We call this sequence data reduction cascade. The FLAMES-UVES pipeline involves the following input data:

- Raw frames:
  - Fibre formatcheck frames to determine guess order and line tables.
  - Calibration fibre frames to determine guess order tables and have a flat field for that fibre.
  - Bias frames to determine a master bias.
  - Dark frames to determine a master dark.
  - Slit flat frames to determine a set of master slit flats covering all the nine fibres.
  - Odd-even-all fibre order definition frames to determine the fibre order traces tables and construct a reference base of non contaminated fibre frames.
  - Fibre arc lamp frames to determine the line table for all fibres.
- calibration data products:
  - Calibration guess line table to have an automatic, accurate and stable wavelength calibration.
  - Calibration fibre guess order table to have a stable order tracing.
  - Calibration fibre order flat frame to have a flat frame to flat the calibration fibre and a base fibre to estimate the contamination of that fibre to the others.

- Master bias, master dark to subtract them from master flat, and science frames.
- Master slit flat fields to correct for different detector pixel efficiencies, the blaze function, the detector fringing at longer wavelengths.
- Line tables to calibrate the object and sky spectra in wavelength.

- Reference files:
  - Line table to produce guess and final line tables.

9.3 Reduction cascade

Here we outline the logical sequence of steps which are needed to perform a complete spectral extraction of FLAMES-UVES data. All these operations can be carried out using commands in the FLAMES context. Examples of data reduction are in chapter 5.

1. For most frames (fibre and slit flat-fields, science frames) a variance frame and a bad-pixel mask must be created.
2. First, a so-called format-check frame is examined. This frame must have been taken with a single fibre illuminated by a (Th-Ar) wavelength-calibration lamp, and all other fibres dark. This frame is compared with a physical model of the CCD illumination, with the help of an appropriate table of lines emitted by the calibration lamp, to derive a first guess of the order positions and of the wavelength-calibration solution.
3. The whole set of fibres is considered (or at least those used in the science exposure), by processing both an odd-numbered- and an even-numbered fibre flat-field frames. In this way a complete (raw) fibre-order position table is obtained.
4. From the latter, the inter-order (background) table is computed, for later computation and subtraction of scattered-light contamination.
5. Slit flat-field frames are processed. At least two half-slit flat-fields are needed to cover the fibre-illuminated region without overlapping, and these are combined and normalized. They will be used to remove pixel-to-pixel effects later. Associated variance frames and bad-pixel masks are also created.
6. odd/even fibre flat-fields are processed. These frames contain the required information on the fibre cross-dispersion profile, and need to be input as separate odd/even fibre frames since adjacent fibres have partially overlapping profiles. Fibres for which there is no corresponding slit-flat-field information are ignored. The frames are corrected for background light, and associated variance frames and bad-pixel masks are created.
7. The algorithm requires also an all-fibre flat-field frame, to compute the relative throughput among the odd and even fibres. It is treated as if it were a science frame, with the same extraction procedure, to obtain for each fibre and order a wavelength-dependent normalization (instead of a spectrum). Since the FLAMES context includes two extraction methods (standard and optimal), the same choice is possible to extract the fibre normalizations here. Moreover, the all-fibre flat-field frame is also used as a reference for the positions of all fibres, since they are simultaneously lit here. For this purpose, the cross-dispersion shifts

between the fibre positions in the odd and even fibre flat-field frames are computed with this common reference, and this additional information is stored in the order-fibre table.

8. Wavelength calibration is done, using a fibre frame with each fibre illuminated by a ThAr lamp, the guess line and fibre-order tables determined from previous data reduction steps. This step extracts each fibre spectrum, looks for ThAr lines and identifies them with help of a guess solution.

9. Eventually, the science spectra are extracted, either using standard or optimal extraction. Because of UVES stability limitations, there may be shifts between the fibre images in the flat-fields and in the science frames. These shifts are expected to occur in the cross-dispersion direction and to be at most ± 1 pixel. Since especially optimal extraction is extremely sensitive to such shifts (spectra with large S/N cannot be extracted satisfactorily for shifts larger than 0.1 pix), the extraction routine computes the value of the shift (using a correlation-function method), and applies the opposite shift to the fibre flat-fields (that's why pixel-to-pixel effects need to be computed separately using slit flat-fields). After doing that, the optimal extraction is performed using the fibre flat-fields as a model to fit the science frame at every wavelength (the fit amplitudes are the extracted spectra at each wavelength), using at the same time the fibre profile information to deconvolve the partial fibre overlap. Standard extraction is somewhat simpler, doing only a sum of the science frame fibre spectra over a pre-defined window across dispersion, without fitting, but including also a deconvolution of fibre cross-contamination.

10. Next, extracted spectra are wavelength-calibrated.

11. Last, wavelength-calibrated, merged spectra are created.

9.3.1 Data reduction peculiarities

The data reduction software design to reduce FLAMES-UVES fibre-echelle mode data (described in [2], [3] and [5]) has some data reduction peculiarities for example with respect to the standard echelle data reduction, or with respect to the extraction of fibre spectra designed for FEROS.

With respect to a standard echelle package:

- The fibre flat field spectra are used as "true" physical model of the light distribution on the detector in the optimal extraction, instead of some analytic approximation (e.g. as in UVES-echelle mode where a Gaussian+constant function distribution is used).
- A very good solution of the adjacent fibre contamination (which is a peculiar problem of fibre-fed multi-object spectrographs with close packed fibres and thus does not occur in echelle mode) is achieved.
- Spectra corresponding to different fibres, having a different fibre throughput, are corrected giving reliable relative fibre throughputs.
- Thanks to the use of three different kind of flat-field spectra (single fibre, odd-even-all fibres, slit flat fields) usually the final merged spectra have a very high quality, in which any residual oscillation artifacts (still present in some high S/N UVES echelle extracted spectra) are well below the noise level.

With respect to FEROS we have the same difference as before with an additional one:

- In FEROS the flat fielding is done on the extracted spectra, extracting before the science and the flat field and dividing the first by the latter. Although this procedure has empirically been shown to lead to negligible differences with respect to the "correct" one, we still use the "correct" one (first correct science by flat field, next extract the spectra).

The adopted data reduction together with to the mentioned pros implies a number of contra. Essentially we have two. We need many calibrations, for example comparing, many more than, for example, UVES operated in echelle mode.

The achieved very good extraction quality is paid with a quite intensive computational cost. On Paranal the hardware dedicated to data reduction in support of FLAMES-UVES changed from the probably more reliable (and expensive) HP-UX architecture to the faster PC-Linux architecture. To improve even more computational performance we had to define additional simplified (and with lower quality products) data reduction modes (“fast” and “quick”) just in support of operations even if at the moment they are not used.

10 Pipeline Recipes Interfaces

In this section we provide for each recipe examples the required input data (and their classification tags). In the following we assume that the input files (in our examples raw data are located in the directory /path_raw, reference data are in the directory /path_ref and pipeline products are in the directory /path_pro) are existing FITS files (e.g. /data1/uves/com2/UVES.2004-08-16T02:54:04.353.fits and /cal/uves/ech/cal/thargood_3.fits, /cal/uves/ech/cal/linetable_redl.fits, /cal/uves/ech/cal/linetable_redu.fits).

We also provide a full list of pipeline products for each recipe, indicating their default recipe name (optionally replaced by EsoRex to a given standard), the value of the FITS keyword HIERARCH ESO PRO CATG (in short PRO.CATG) and a short description. The relevant keywords are PRO.CATG, used to classify each frame, and to associate to each raw frame the proper calibration frame.

The data may be recognized and organized according to the values of the following FITS keywords:

Association keyword	Information
HIERARCH ESO DPR TYPE	raw data type
HIERARCH ESO DPR CATG	raw data category
HIERARCH ESO DPR TECH	raw data technique
HIERARCH ESO INS GRATj WLEN	Instrument setup central wavelength
HIERARCH ESO DET DIT	Integration time

The pipeline is able to also process pipeline products generated by the MIDAS based pipeline. The viceversa is not true.

For each recipe we also list the input parameters (as they appear in the recipe configuration file), the corresponding parameter aliases (to be set on the command line) and their default values. Each recipe has the following common parameters:

parameter	alias	default
debug	debug	FALSE
plotter	plotter	no
process_chip	process_chip	both

The parameter **debug** may be set to TRUE to generate a more detailed recipe log and extra products for debugging purposes. The parameter **plotter** may be set to 'gnuplot -persist' in order to generate some plots. The parameter **process_chip** may be set, in case of RED arm data, to 'redl' or 'redu', to reduce only the corresponding chip data. This allows to use different parameter values to reduce data of the two UVES red arm detector chips.

A full description of each parameter is obtainable by running the command **esorex --parameters**, or **esorex --help** or **esorex --man-page**, or by looking at the Recipe Input Parameters section of the dedicated Gasgano window. Also, the role of the most relevant parameters is described in section 11.

Also quality control parameters are given. Those are stored in the corresponding pipeline products. More and updated information on instrument quality control can be found on `www.eso.org/qc`.

10.1 flames_cal_predict

The recipe flames_cal_predict computes the guess line and order tables using a model of UVES and the information on the atmospheric pressure, temperature and the instrument setting provided by the FITS header of the raw formatcheck frame.

10.1.1 Input

```
/path_raw/flames_uves_arc_lamp_form_red.fits FIB_ARC_LAMP_FORM_RED
/path_ref/thargood_3.fits                    LINE_REFER_TABLE
```

10.1.2 Output

default recipe file name	format	PRO.CATG	short description
orderguesstable_redl.fits	table	FIB_ORD_GUE_TAB_REDL	Guess order table
lineguesstable_redl.fits	table	FIB_LIN_GUE_TAB_REDL	Guess line table
orderguesstable_redu.fits	table	FIB_ORD_GUE_TAB_REDU	Guess order table
lineguesstable_redu.fits	table	FIB_LIN_GUE_TAB_REDU	Guess line table

The guess line table contains the following columns:

X	Position along x
Y	Position along y
PEAK	line peak
Ident	line catalog wavelength
YNEW	Computed predicted line y position
Order	Relative order number
WAVEC	Predicted line wavelength of line peak
Aux	Product of wavelength and order number
XREG	
Pixel	Local dispersion
RORD	
XPRED	Predicted X line position
YPRED	Predicted Y line position
XDIF	Difference between measured and predicted X line position
YDIF	Difference between measured and predicted Y line position
SELPLOT	selection column

The guess order table contains the following columns:

ABS_ORDER	Absolute order number
ORDER	Relative order number
X	Position along x
Y	Position along y
YFIT	Computed predicted order y position
RESIDUAL	Residual (Y-YFIT)

10.1.3 Quality control

The pipeline generates the following QC parameters:

QC.MODEL.NLINALL	Total number of detected lines
QC.MODEL.NLINSEL	Number of selected lines
QC.MODEL.DIFFXRMS	RMS difference of predicted and measured line x positions
QC.MODEL.DIFFXAVG	Mean difference of predicted and measured line x positions
QC.MODEL.DIFFXMED	Median difference of predicted and measured line x positions
QC.MODEL.DIFFYRMS	RMS difference of predicted and measured line y positions
QC.MODEL.DIFFYAVG	Mean difference of predicted and measured line y positions
QC.MODEL.DIFFYMED	Median difference of predicted and measured line y positions
QC.MODEL.RESXRMS	Std dev of X difference to physical model
QC.MODEL.RESXAVG	Average of X difference to physical model
QC.MODEL.RESXMED	Median of X difference to physical model
QC.MODEL.RESYRMS	Std dev of Y difference to physical model
QC.MODEL.RESYAVG	Average of Y difference to physical model
QC.MODEL.RESYMED	Median of Y difference to physical model
QC.MODEL.WLENMIN	Minimum predicted lines wavelength
QC.MODEL.WLENMAX	Maximum predicted lines wavelength
QC.MODEL.ORDMIN	Minimum predicted absolute order
QC.MODEL.ORDMAX	Maximum predicted absolute order
QC.WLENMIN	Minimum wavelength of spectral format
QC.WLENMAX	Maximum wavelength of spectral format
QC.ORDMIN	Minimum relative order
QC.ORDMAX	Maximum relative order
QC.FIB1.ABSTRANS	Average transmission countrate measured on each fibre
QC.FIB1.NHOTPIX	Number of found hot pixels and hot columns.
QC.FIB1.PLATENO	This is the Id of the plate to which the fibres are connected to

10.1.4 Parameters

parameter	alias	default
uves_cal_predict.mbox_x	mbox_x	40
uves_cal_predict.mbox_y	mbox_y	40
uves_cal_predict.trans_x	trans_x	0.0
uves_cal_predict.trans_y	trans_y	0.0
uves_cal_predict.ech_angle_off	ech_angle_off	0.0
uves_cal_predict.cd_angle_off	cd_angle_off	0.0
uves_cal_predict.rot_angle_off	ccd_rot_angle_off	0.0
uves_cal_predict.compute_regression_sw	compute_regression_sw	TRUE
uves_cal_predict.x_axis_scale	x_axis_scale	0.0
uves_cal_predict.y_axis_scale	y_axis_scale	0.0
uves_cal_predict.def_pol1	def_pol1	4
uves_cal_predict.def_pol2	def_pol2	5

uves_cal_predict.kappa	kappa	4.5
uves_cal_predict.tol	tol	2.0

10.2 flames_cal_orderpos

The recipe flames_cal_orderpos generates an order table from a set of raw frames taken with the calibration fibre illuminated by a continuum lamp.

10.2.1 Input

```
/path_raw/flames_uves_order_flat_red.fits FIB_ORDEF_RED
```

10.2.2 Output

default recipe file name	format	PRO.CATG	short description
ordertable_redl.fits	table	FIB_ORD_TAB_REDL	order table
order_def_redl.fits	table	FIB_ORDEF_REDL	fiber order def rotated frame
ordertable_redu.fits	table	FIB_ORD_TAB_REDU	order table
order_def_redu.fits	table	FIB_ORDEF_REDU	fiber order def rotated frame

The output table contains the columns

Order	Relative order number
X	Position along x
Y	Order line centroid location
dY	Uncertainty of Y
Residual_Square	Squared residual
OrderRMS	Root mean squared residual of initial one-dimensional linear fit of order
OrderSlope	Slope of order
Yfit	The fitted order location
dYfit_Square	Variance of Yfit
Residual	(Y - Yfit)

10.2.3 Quality control

QC.ORD.RESIDMIN	Min residuals in order table
QC.ORD.RESIDMAX	Max residuals in order table
QC.ORD.RESIDAVG	Mean residuals in order table
QC.ORD.RESIDRMS	RMS residuals in order table
QC.ORD.NPRED	Predicted number of orders
QC.ORD.NDET	Detected number of orders

QC.ORD.NPOSALL	Number of positions found
QC.ORD.NPOSSEL	Number of positions selected
QC.ORDMIN	Minimum (relative) order value
QC.ORDMAX	Maximum (relative) order value

Where the residuals measure the difference between the order solution obtained by applying the polynomial model and the corresponding order location measurements on the frame.

10.2.4 Parameters

parameter	alias	default
uves_cal_orderpos.preproc.use_guess_tab	use_guess_tab	1
uves_cal_orderpos.preproc.radx	radx	2
uves_cal_orderpos.preproc.rady	rady	1
uves_cal_orderpos.preproc.mmethod	mmethod	median
uves_cal_orderpos.preproc.backsubgrid	backsubgrid	50
uves_cal_orderpos.preproc.backsubradiusy	backsubradiusy	2
uves_cal_orderpos.preproc.backsubkappa	backsubkappa	4.0
uves_cal_orderpos.preproc.backsubdegx	backsubdegx	2
uves_cal_orderpos.preproc.backsubdegy	backsubdegy	2
uves_cal_orderpos.hough.samplewidth	samplewidth	50
uves_cal_orderpos.hough.minslope	minslope	0.0
uves_cal_orderpos.hough.maxslope	maxslope	0.2
uves_cal_orderpos.hough.sloperes	sloperes	120
uves_cal_orderpos.hough.norders	norders	0
uves_cal_orderpos.hough.pthres	pthres	0.2
uves_cal_orderpos.trace.tracestep	tracestep	10
uves_cal_orderpos.trace.maxthresh	maxthresh	0.6
uves_cal_orderpos.trace.minthresh	minthresh	0.2
uves_cal_orderpos.trace.lowerthresh	lowerthresh	0.75
uves_cal_orderpos.trace.traceiter	traceiter	3
uves_cal_orderpos.trace.tracelength	tracelength	0.9
uves_cal_orderpos.trace.maxgap	maxgap	0.2
uves_cal_orderpos.reject.maxrms	maxrms	100.0
uves_cal_orderpos.reject.defpol1	defpol1	-1
uves_cal_orderpos.reject.defpol2	defpol2	-1
uves_cal_orderpos.reject.kappa	kappa	6.0

10.3 uves_cal_mbias

The recipe uves_cal_mbias computes a master bias frame.

10.3.1 Input

```
/path_raw/uves_bias_red1.fits  BIAS_RED
/path_raw/uves_bias_red2.fits  BIAS_RED
/path_raw/uves_bias_red3.fits  BIAS_RED
/path_raw/uves_bias_red4.fits  BIAS_RED
/path_raw/uves_bias_red5.fits  BIAS_RED
```

10.3.2 Output

default recipe file name	format	PRO.CATG	short description
masterbias_red.fits	2d (pix-pix) image	MASTER_BIAS_RED	master bias frame

10.3.3 Quality control

The recipe computes the following QC parameters:

QC.DUTYCYCL	Time to store a frame
QC.OUT1.RON.RAW	Read noise frame in ADU
QC.OUT1.RON.MASTER	Read noise frame in ADU
QC.OUT1.STRUCTY	Structure in Y
QC.OUT1.STRUCTX	Structure in X

10.3.4 Parameters

There are no recipe specific input parameter for uves_cal_mbias.

parameter	alias	default
uves_cal_mbias.clean_traps	clean_traps	TRUE

10.4 uves_cal_mdark

The recipe uves_cal_mdark computes a master dark frame.

10.4.1 Input

```
/path_raw/uves_dark_red1.fits  DARK_RED
/path_raw/uves_dark_red2.fits  DARK_RED
/path_raw/uves_dark_red3.fits  DARK_RED
/path_raw/masterbias_redl.fits  MASTER_BIAS_REDL
/path_raw/masterbias_redu.fits  MASTER_BIAS_REDU
```

10.4.2 Output

default recipe file name	format	PRO.CATG	short description
masterdark_redl.fits	2d (pix-pix) image	MASTER_DARK_REDL	master dark frame
masterdark_redu.fits	2d (pix-pix) image	MASTER_DARK_REDU	master dark frame

10.4.3 Quality control

The pipeline monitors the number of coadded frames (PRO.DATANCOM). DFO trend the dark median level (PRO.DATAMED).

QC.DATANCOM	Number of coadded frames
PRO.DATAMED	Median frame level
QC.REGij.MIN	Min of region i j of size **box_sx** $\times$ **box_sy**
QC.REGij.MAX	Max of region i j of size **box_sx** $\times$ **box_sy**
QC.REGij.AVG	Mean of region i j of size **box_sx** $\times$ **box_sy**
QC.REGij.MED	Median of region i j of size **box_sx** $\times$ **box_sy**
QC.REGij.RMS	Rms of region i j of size **box_sx** $\times$ **box_sy**
QC.REG.MIN.MIN	Min of all region Mins
QC.REG.MIN.MAX	Max of all region Mins
QC.REG.MIN.AVG	Mean of all region Mins
QC.REG.MIN.MED	Median of all region Mins
QC.REG.MIN.RMS	Rms of all region Mins
QC.REG.MAX.MIN	Min of all region Maxs
QC.REG.MAX.MAX	Max of all region Maxs
QC.REG.MAX.AVG	Mean of all region Maxs
QC.REG.MAX.MED	Median of all region Maxs
QC.REG.MAX.RMS	Rms of all region Maxs
QC.REG.AVG.MIN	Min of all region Means
QC.REG.AVG.MAX	Max of all region Means
QC.REG.AVG.AVG	Mean of all region Means
QC.REG.AVG.MED	Median of all region Means
QC.REG.AVG.RMS	Rms of all region Means
QC.REG.MED.MIN	Min of all region Medians
QC.REG.MED.MAX	Max of all region Medians
QC.REG.MED.AVG	Mean of all region Medians
QC.REG.MED.MED	Median of all region Medians
QC.REG.MED.RMS	Rms of all region Medians
QC.REG.RMS.MIN	Min of all region Rms
QC.REG.RMS.MAX	Max of all region Rms
QC.REG.RMS.AVG	Mean of all region Rms
QC.REG.RMS.MED	Median of all region Rms
QC.REG.RMS.RMS	Rms of all region Rms

10.4.4 Parameters

parameter	alias	default
uves_cal_mdark.reg.num_x	num_x	4
uves_cal_mdark.reg.num_y	num_y	4
uves_cal_mdark.reg.box_sx	box_sx	100
uves_cal_mdark.reg.box_sy	box_sy	100
uves_cal_mdark.reg.border_sx	border_sx	100
uves_cal_mdark.reg.border_sy	border_sy	100
uves_cal_mdark.reg.when	when	0

10.5 flames_cal_mkmaster

The recipe flames_cal_mkmaster computes the master slit flat frames.

10.5.1 Input

```
/path_raw/uves_flat_set1_red1.fits FLAT_RED
/path_raw/uves_flat_set1_red2.fits FLAT_RED
/path_raw/uves_flat_set1_red3.fits FLAT_RED
/path_raw/uves_flat_set2_red1.fits FLAT_RED
/path_raw/uves_flat_set2_red2.fits FLAT_RED
/path_raw/uves_flat_set2_red3.fits FLAT_RED
/path_raw/uves_flat_set3_red1.fits FLAT_RED
/path_raw/uves_flat_set3_red2.fits FLAT_RED
/path_raw/uves_flat_set3_red3.fits FLAT_RED

/path_pro/masterbias_redl.fits MASTER_BIAS_REDL
/path_pro/ordertable_redl.fits ORDER_TABLE_REDL
/path_pro/masterbias_redu.fits MASTER_BIAS_REDU
/path_pro/ordertable_redu.fits ORDER_TABLE_REDU
```

10.5.2 Output

default recipe file name	format	PRO.CATG	short description
masterflat_set1_redl.fits	2d (pix-pix) image	MASTER_SFLAT_REDL	master flat frame
masterflat_bkg_set1_redl.fits	2d (pix-pix) image	BKG_SFLAT_REDL	background of master flat frame
masterflat_set2_redl.fits	2d (pix-pix) image	MASTER_SFLAT_REDL	master flat frame
masterflat_bkg_set2_redl.fits	2d (pix-pix) image	BKG_SFLAT_REDL	background of master flat frame
masterflat_set3_redl.fits	2d (pix-pix) image	MASTER_SFLAT_REDL	master flat frame
masterflat_bkg_set3_redl.fits	2d (pix-pix) image	BKG_SFLAT_REDL	background of master flat frame
masterflat_set1_redu.fits	2d (pix-pix) image	MASTER_SFLAT_REDU	master flat frame
masterflat_bkg_set1_redu.fits	2d (pix-pix) image	BKG_SFLAT_REDU	background of master flat frame

masterflat_set2_redu.fits	2d (pix-pix) image	MASTER_SFLAT_REDU	master flat frame
masterflat_bkg_set2_redu.fits	2d (pix-pix) image	BKG_SFLAT_REDU	background of master flat frame
masterflat_set3_redu.fits	2d (pix-pix) image	MASTER_SFLAT_REDU	master flat frame
masterflat_bkg_set3_redu.fits	2d (pix-pix) image	BKG_SFLAT_REDU	background of master flat frame

10.5.3 Quality control

The pipeline monitors the number of coadded frames (PRO.DATANCOM). Additional quality control information is monitored by DFO and can be found at `www.eso.org/qc`.

10.5.4 Parameters

parameter	alias	default
uves_cal_mflat.backsub.mmethod	backsub.mmethod	median
uves_cal_mflat.backsub.npoints	backsub.npoints	82
uves_cal_mflat.backsub.radiusy	backsub.radiusy	2
uves_cal_mflat.backsub.sdegree	backsub.sdegree	1
uves_cal_mflat.backsub.smoothx	backsub.smoothx	-1.0
uves_cal_mflat.backsub.smoothy	backsub.smoothy	-1.0

10.6 flames_cal_prep_sff_ofpos

The recipe uves_cal_prep_sff_ofpos is used to determine the fibre order table and construct several frames needed to extract a science fibre frame.

10.6.1 Input

```
/path_raw/flames_uves_odd_red.fits   FIB_FF_ODD_RED
/path_raw/flames_uves_even_red.fits  FIB_FF_EVEN_RED
/path_raw/flames_uves_all_red.fits   FIB_FF_ALL_RED
/path_pro/masterbias_redl.fits       MASTER_BIAS_REDL
/path_pro/ordertable_redl.fits       FIB_ORD_GUE_REDL
/path_pro/masterflat_set1_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterflat_set2_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterflat_set2_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterbias_redu.fits       MASTER_BIAS_REDU
/path_pro/ordertable_redu.fits       FIB_ORD_GUE_REDU
/path_pro/masterflat_set1_redu.fits  MASTER_SFLAT_REDU
/path_pro/masterflat_set2_redu.fits  MASTER_SFLAT_REDU
/path_pro/masterflat_set2_redu.fits  MASTER_SFLAT_REDU
```

10.6.2 Output

default recipe filename	format	PRO.CATG	short description
xt_odd_l	table	FIB_FF_ODD_INFO_TAB	info table
xt_even_l	table	FIB_FF_EVEN_INFO_TAB	info table
xt_all_l	table	FIB_FF_ODD_INFO_TAB	info table
slitff_common_redl	cube	SLIT_FF_COM_REDL	slitff common frame
slitff_norm_redl	image	SLIT_FF_NOR_REDL	slitff common frame
slitff_dtc_redl	cube	SLIT_FF_DTC_REDL	slitff data frame
slitff_bpc_redl	cube	SLIT_FF_BPC_REDL	slitff badpixel frame
slitff_sgc_redl	cube	SLIT_FF_SGC_REDL	slitff sigma frame
slitff_bnc_redl	iper-cube	SLIT_FF_BNC_REDL	slitff bound frame
fibff_common_redl	iper-cube	FIB_FF_COM_REDL	fibff common frame
fibff_norm_redl	cube	FIB_FF_NOR_REDL	fibff common frame
fibff_dtc_redl	cube	FIB_FF_DTC_REDL	fibff data frame
fibff_bpc_redl	cube	FIB_FF_BPC_REDL	fibff badpixel frame
fibff_sgc_redl	cube	FIB_FF_SGC_REDL	fibff sigma frame
fib_ordef_redl	table	FIB_ORDEF_TABLE_REDL	fibre order table
slitff_common_redu	cube	SLIT_FF_COM_REDU	slitff common frame
slitff_norm_redu	image	SLIT_FF_NOR_REDU	slitff common frame
slitff_dtc_redu	cube	SLIT_FF_DTC_REDU	slitff data frame
slitff_bpc_redu	cube	SLIT_FF_BPC_REDU	slitff badpixel frame
slitff_sgc_redu	cube	SLIT_FF_SGC_REDU	slitff sigma frame
slitff_bnc_redu	iper-cube	SLIT_FF_BNC_REDU	slitff bound frame
fibff_common_redu	iper-cube	FIB_FF_COM_REDU	fibff common frame
fibff_norm_redu	cube	FIB_FF_NOR_REDU	fibff common frame
fibff_dtc_redu	cube	FIB_FF_DTC_REDU	fibff data frame
fibff_bpc_redu	cube	FIB_FF_BPC_REDU	fibff badpixel frame
fibff_sgc_redu	cube	FIB_FF_SGC_REDU	fibff sigma frame
fib_ordef_redu	table	FIB_ORDEF_TABLE_REDU	fibre order table

10.6.3 Quality control

10.6.4 Parameters

parameter	alias	default
flames_cal_prep_sff_ofpos.ext_method	ext_method	opt
flames_cal_prep_sff_ofpos.bias_method	bias_method	M
flames_cal_prep_sff_ofpos.bias_value	bias_value	200
flames_cal_prep_sff_ofpos.filter_switch	filter_switch	none
flames_cal_prep_sff_ofpos.sat_thr	sat_thr	55000
flames_cal_prep_sff_ofpos.fileprep	fileprep	TRUE
flames_cal_prep_sff_ofpos.cubify	cubify	TRUE

10.7 flames_cal_wavecal

The recipe flames_cal_wavecal is used to determine the wavelength dispersion coefficients and construct a wavelength calibration table.

10.7.1 Input

```
/path_raw/flames_uves_arc_lamp_red.fits    FIB_ARC_LAMP_RED
/path_pro/ordertable_redl.fits      FIB_ORDEF_TABLE_REDL
/path_pro/lineguesstable_redl.fits  FIB_LIN_GUE_REDL
/path_pro/gue580o1re1x1.fits        FIB_LIN_GUE_REDL
/path_pro/ordertable_redu.fits      FIB_ORDEF_TABLE_REDU
/path_pro/lineguesstable_redu.fits  FIB_LIN_GUE_REDU
/path_pro/gue580o1rm1x1.fits        FIB_LIN_GUE_REDU
/path_ref/thargood_3.fits           LINE_REFER_TABLE
```

10.7.2 Output

default recipe file name	format	PRO.CATG	short description
linetable_redl.fits	table	FIB_LINE_TABLE_REDL	line table
linetable_redu.fits	table	FIB_LINE_TABLE_REDU	line table

The output line table(s), LINE_TABLE_REDL, LINE_TABLE_REDU contains the columns

X	Horizontal position (from Gaussian fit) of detected line
dX	Uncertainty (one sigma) of X
XWidth	Width (in pixels) of detected line (from Gaussian fit)
Y	Relative order number of detected line
Peak	Intensity of detected line
Background	Fitted background (ADU) of detected line
Slope	Linear background slope (ADU/pixel) of detected line
Order	Relative order number of detected line
AbsOrder	Absolute order number of detected line
Ynew	Vertical position of detected line
WaveC	Wavelength of this line (computed using the resulting dispersion relation)
dLambdaC	Uncertainty (one sigma) of 'WaveC'.
Pixel	The local dispersion in A/pixel.
Residual	Residual (in A) of this line
Residual_pix	Residual (in pixels) of this line
Lambda_candidate	Nearest line in catalogue
dLambda_cat_sq	Squared distance to nearest catalogue line
dLambda_nn_sq	Squared distance to nearest neighbour multiplied by ALPHA
Ident	The wavelength associated with this emission line,

	or NULL if this line was not identified.
dIdent	Uncertainty of catalogue wavelength
Select	1 if the line was identified, 0 otherwise
NLinSol	1 if the line was identified and accepted for the polynomial fit, 0 otherwise
Intensity	Intensity of detected line scaled to unit exposure time. (This column is present only if a LINE_INTMON_TABLE is provided.)

The 2nd table extension contains the dispersion relation on the form $p(x,m) = \lambda \cdot m$, where m is the order number. The 3rd table extension contains the map from (pixel, pixel)-space to physical order numbers (used internally by the calibration recipe, a 2d polynomial on the form $p(x,y) = m$).

If there is more than one extraction window (default is 3), the results of each calibration is stored in subsequent table extensions of the same FITS file. For example, extensions 4, 5 and 6 would contain the resulting line table (and its two associated polynomials) for the second extraction window. The results for the calibration of the n'th extraction window is stored in extensions (3*n - 2) to 3*n.

The offset of the extraction window is stored in the FITS keyword like "HISTORY OFFSET -7.5". The corresponding window number (counting from 1) is stored in keywords like "HISTORY WINDOW 3"

The polynomials are stored in table extensions as in the example

Order1	Order2	Coeff
-1	-1	a0
-1	-1	a1
-1	-1	a2
-1	-1	b0
-1	-1	b1
-1	-1	b2
0	0	c00
0	1	c01
0	2	c02
1	0	c10
1	1	c11
1	2	c12
2	0	c20
2	1	c21
2	2	c22

The third column contains the polynomial coefficients corresponding to the degree defined by the two first columns. The six first table rows defines a linear transformation of the dependent and independent variables. For example the table shown represents the polynomial p defined by

$$(p(x,y) - a_0)/b_0 = q((x - a_1)/b_1, (y - a_2)/b_2)$$

and

$$q(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} c_{ij} x^i y^j$$

The linear transformation of the three variables was introduced in order to ensure numerical stability in the polynomial fitting routine.

10.7.3 Quality control

The recipe computes the following quality control parameters:

QC.FWHMAVG	Average FWHM of lines selected
QC.FWHMRMS	Standard deviation of FWHM of selected lines
QC.FWHMMED	Median FWHM of selected lines
QC.RESOLAVG	Average resolving power of selected lines
QC.RESOLRMS	Standard deviation of the resolving power of selected lines
QC.RESOLMED	Median resolving power of selected lines
QC.LINE.RESIDAVG	Mean of residuals of line positions to fit
QC.LINE.RESIDRMS	Sigma of residuals of line positions to fit
QC.WLENMIN	Minimum wavelength of detected order
QC.WLENMAX	Maximum wavelength of detected order
QC.ORDMIN	Minimum order number detected
QC.ORDMAX	Maximum order number detected
QC.NLINTOT	Total number of lines found on the frame
QC.NLINSEL	Number of selected lines
QC.NLINRES1	Number of lines with residuals < 0.1 nm
INS.SLIT3.WID	
INS.GRAT2.WLEN	
INS.TEMP2.MEAN	
QC.NHOTPIX	
QC.PLATENO	
QC.FIB8.DRSNO	8
QC.FIB8.SEQ	8
QC.FIB8.POS	23.3773
QC.FIB8.MSK	1
QC.FIB8.FWHMAVG	5.28
QC.FIB8.FWHMRMS	0.7600
QC.FIB8.FWHMMED	5.2542
QC.FIB8.RESOLAVG	48299.3577
QC.FIB8.RESOLRMS	4996.5595
QC.FIB8.RESOLMED	48972.5171
QC.FIB8.LINE.RESIDAVG	-0.0000
QC.FIB8.LINE.RESIDRMS	0.1596

QC.FIB8.WLENMIN	475.6704
QC.FIB8.WLENMAX	576.3529
QC.FIB8.ORDMIN	106
QC.FIB8.ORDMAX	128
QC.FIB8.NLINTOT	1535
QC.FIB8.NLINSEL	676
QC.FIB8.NLINSOL	572
QC.FIB8.NLINRES1	676

The line FWHMs and the corresponding resolving power are measured along the dispersion direction (see also Figure **??**). The residuals measures the differences between the solution from the polynomial fit model and the corresponding line positions found on the arc lamp frame (see also Figure **??**).

10.7.4 Parameters

parameter	alias	default
flames_cal_wavecal.nwindows	nwindows	3
flames_cal_wavecal.length	length	-1
flames_cal_wavecal.offset	offset	0.0
flames_cal_wavecal.extract.method	extract.method	average
flames_cal_wavecal.extract.kappa	extract.kappa	10.0
flames_cal_wavecal.extract.chunk	extract.chunk	32
flames_cal_wavecal.extract.profile	extract.profile	auto
flames_cal_wavecal.extract.skymethod	extract.skymethod	optimal
flames_cal_wavecal.extract.oversample	extract.oversample	5
flames_cal_wavecal.extract.best	extract.best	TRUE
flames_cal_wavecal.search.range	range	8
flames_cal_wavecal.search.minlines	minlines	0
flames_cal_wavecal.search.maxlines	maxlines	0
flames_cal_wavecal.search.centeringmethod	centeringmethod	gaussian
flames_cal_wavecal.first.shiftmax	shiftmax	10.0
flames_cal_wavecal.first.shiftstep	shiftstep	0.1
flames_cal_wavecal.first.shifttoler	shifttoler	0.05
flames_cal_wavecal.identify.alpha	alpha	0.1
flames_cal_wavecal.identify.maxerror	maxerror	20.0
flames_cal_wavecal.identify.degree	degree	4
flames_cal_wavecal.calibrate.tolerance	tolerance	0.6
flames_cal_wavecal.rebin.wavestep	rebin.wavestep	-1.0
flames_cal_wavecal.rebin.scale	rebin.scale	FALSE

10.8 flames_obs_scired

This recipe reduces a science fiber frame.

10.8.1 Input

```
/path_raw/flames_uves_science_red.fits FIB_SCI_RED
/path_pro/slitff_com_redl.fits         SLIT_FF_COM_REDL
/path_pro/slitff_nor_redl.fits         SLIT_FF_NOR_REDL
/path_pro/slitff_dtc_redl.fits         SLIT_FF_DTC_REDL
/path_pro/slitff_bpc_redl.fits         SLIT_FF_BPC_REDL
/path_pro/slitff_sgc_redl.fits         SLIT_FF_SGC_REDL
/path_pro/fibff_com_redl.fits          FIB_FF_COM_REDL
/path_pro/fibff_nor_redl.fits          FIB_FF_NOR_REDL
/path_pro/fibff_dtc_redl.fits          FIB_FF_DTC_REDL
/path_pro/fibff_bpc_redl.fits          FIB_FF_BPC_REDL
/path_pro/fibff_sgc_redl.fits          FIB_FF_SGC_REDL
/path_pro/fibff_bnc_redl.fits          FIB_FF_BNC_REDL
/path_pro/fib_ordef_redl.fits          FIB_ORDEF_TABLE_REDL
/path_pro/slitff_com_redu.fits         SLIT_FF_COM_REDU
/path_pro/slitff_nor_redu.fits         SLIT_FF_NOR_REDU
/path_pro/slitff_dtc_redu.fits         SLIT_FF_DTC_REDU
/path_pro/slitff_bpc_redu.fits         SLIT_FF_BPC_REDU
/path_pro/slitff_sgc_redu.fits         SLIT_FF_SGC_REDU
/path_pro/fibff_com_redu.fits          FIB_FF_COM_REDU
/path_pro/fibff_nor_redu.fits          FIB_FF_NOR_REDU
/path_pro/fibff_dtc_redu.fits          FIB_FF_DTC_REDU
/path_pro/fibff_bpc_redu.fits          FIB_FF_BPC_REDU
/path_pro/fibff_sgc_redu.fits          FIB_FF_SGC_REDU
/path_pro/fibff_bnc_redu.fits          FIB_FF_BNC_REDU
/path_pro/fib_ordef_redu.fits          FIB_ORDEF_TABLE_REDU
```

10.8.2 Output

default recipe file name	format	PRO.CATG	short description
fxb_l_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDL	extracted, flatfielded raw frame
wfxb_l_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDL	rebinned, extracted, flatfielded raw frame
fxb_l_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDL	error frame
wfxb_l_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDL	error frame
mwfxb_l_rawsig000i	1d (wav) image	MWXB_SCI_RAW_REDL	merged, rebinned, extracted, flat fielded raw frame
fxb_l_000i	2d (pix-ord) image	XB_SCI_REDL	extracted, flatfielded raw frame
wfxb_l_000i	2d (wav-ord) image	WXB_SCI_REDL	rebinned, extracted, flatfielded raw frame
fxb_l_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDL	error frame
wfxb_l_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDL	error frame
mwfxb_l_sig000i	1d (wav) image	MWXB_SCI_REDL	merged, rebinned, extracted, flat fielded raw frame
fxb_u_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDU	extracted, flatfielded raw frame
wfxb_u_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDU	rebinned, extracted, flatfielded raw frame
fxb_u_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDU	error frame
wfxb_u_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDU	error frame
mwfxb_u_rawsig000i	1d (wav) image	MWXB_SCI_RAW_REDU	merged, rebinned, extracted, flat fielded raw frame
fxb_u_000i	2d (pix-ord) image	XB_SCI_REDU	extracted, flatfielded raw frame
wfxb_u_000i	2d (wav-ord) image	WXB_SCI_REDU	rebinned, extracted, flatfielded raw frame

fxb_u_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDU	error frame
wfxb_u_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDU	error frame
mwfxb_u_sig000i	1d (wav) image	MWXB_SCI_REDU	merged, rebinned, extracted, flat fielded raw frame

10.8.3 Quality control

The recipe computes the folowing quality control parameters in the FITS header of the reduced frame.

10.8.4 Parameters

parameter	alias	default
flames_obs_scired.ext_method	ext_method	opt
flames_obs_scired.cor_max_fnd	cor_max_fnd	Y
flames_obs_scired.cor_def_rng	cor_def_rng	6
flames_obs_scired.cor_def_pnt	cor_def_pnt	25
flames_obs_scired.cor_def_off	cor_def_off	0.0
flames_obs_scired.bias_method	bias_method	-1.0
flames_obs_scired.bias_value	bias_value	optimal
flames_obs_scired.cubify_sw	cubify_sw	10.0
flames_obs_scired.filt_sw	filt_sw	32
flames_obs_scired.bkg_max_io_win	bkg_max_io_win	auto
flames_obs_scired.bkg_xy_win_sz_x	bkg_xy_win_sz_x	optimal
flames_obs_scired.bkg_xy_win_sz_y	bkg_xy_win_sz_y	5
flames_obs_scired.pixel_thresh_max	pixel_thresh_max	TRUE
flames_obs_scired.pixel_thresh_min	pixel_thresh_min	-1.0
flames_obs_scired.input_fmt_cube	input_fmt_cube	TRUE
flames_obs_scired.output_fmt_cube	output_fmt_cube	0.0
flames_obs_scired.drs_k_s_thre	drs_k_s_thre	-1.0
flames_obs_scired.drs_base_name	drs_base_nam	extract
flames_obs_scired.drs_maxyshift	drs_maxyshift	FALSE
flames_obs_scired.drs_ext_w_siz	drs_ext_w_siz	-1.0
flames_obs_scired.reduce.rebin.wavestep	reduce.rebin.wavestep	-1.0
flames_obs_scired.reduce.rebin.scale	reduce.rebin.scale	FALSE
flames_obs_scired.merge	merge	optimal

10.9 flames_obs_redchain

10.9.1 Input

```
/path_raw/uves_bias_red1.fits    BIAS_RED
/path_raw/uves_bias_red2.fits    BIAS_RED
/path_raw/uves_bias_red3.fits    BIAS_RED
/path_raw/uves_bias_red4.fits    BIAS_RED
/path_raw/uves_bias_red5.fits    BIAS_RED
```

```
/path_raw/uves_dark_red1.fits    DARK_RED
/path_raw/uves_dark_red2.fits    DARK_RED
/path_raw/uves_dark_red3.fits    DARK_RED
/path_pro/uves_flat_set1_red1.fits    SFLAT_RED
/path_pro/uves_flat_set1_red2.fits    SFLAT_RED
/path_pro/uves_flat_set1_red3.fits    SFLAT_RED
/path_pro/uves_flat_set2_red1.fits    SFLAT_RED
/path_pro/uves_flat_set2_red2.fits    SFLAT_RED
/path_pro/uves_flat_set2_red3.fits    SFLAT_RED
/path_pro/uves_flat_set3_red1.fits    SFLAT_RED
/path_pro/uves_flat_set3_red2.fits    SFLAT_RED
/path_pro/uves_flat_set3_red3.fits    SFLAT_RED
/path_raw/flames_uves_arc_form_red.fits FIB_ARC_FORM_RED
/path_raw/flames_uves_oflat_red.fits    FIB_ORDER_FLAT_RED
/path_raw/flames_uves_odd_red.fits   FIB_FF_ODD_RED
/path_raw/flames_uves_even_red.fits  FIB_FF_EVEN_RED
/path_raw/flames_uves_all_red.fits   FIB_FF_ALL_RED
/path_raw/flames_uves_arc_lamp_red.fits FIB_ARC_LAMP_RED
/path_raw/flames_uves_science_red.fits  FIB_SCI_RED
/path_ref/thargood_3.fits         LINE_REFER_TABLE
```

10.9.2 Output

This recipe generates all the products described for the previous recipes.

10.9.3 Quality control

This recipe generates all the quality control parameters described for the previous recipes.

10.9.4 Parameters

This recipe is controlled by all the data reduction parameters of the previously described recipes. The additional parameter **scired** (default value is TRUE) switches on/off the execution of the last step (science data reduction).

11 Algorithms and recipe details

In this section we describe the main algorithms implemented in the UVES pipeline recipes. Relevant data reduction parameters are typed in **bold** face. For convenience we omit the common prefix **uves.**

11.1 Algorithms

11.1.1 Error model

The pipeline does full error propagation by using the error propagation formula and making the usual assumption about Gaussian error bars. The initial error bars for the i'th pixel are defined as

$$\sigma_i = \sqrt{g(C_{i,ADU} - b_i) + ron^2 + (g^2 - 1)/12}, \tag{1}$$

where $C_{i,ADU}$ is the flux in analog-to-digital units, b_i is the master bias level and g and ron are the detector gain and read-out noise as defined in the FITS header.

11.1.2 Frames preparation

This data reduction step prepares frames read from disk for usage by the following steps. Its input may be a single frame or a set of frames. For each indicated frame, two new frames are created, a variance frame and a bad-pixel mask frame, with suffixes _sigma.fits and _mask.fits. Their names are written into the FITS keywords **SIGMAFRAME** and **BADPXFRAME** of the input frame.

If the **BADPXFRAME** keyword is defined, it is assumed to contain the name of a generic bad-pixel frame to be used as a first guess (i. e. known detector blemishes). If an input frame contains a **BADPXFRAME** FITS keyword, this is assumed to contain the name of a first guess bad-pixel frame to be used for that specific frame. If both the FITS keywords **BADPXFRAME** and **BADPXFRAME** are defined, the corresponding bad-pixel frames are merged.

Moreover, two additional methods to detect and flag bad pixels are provided: recursive median filtering and clipping of values out of a given validity interval.

If recursive median filtering is enabled setting the parameter **filter_switch** to median, for each pixel the median of the good pixels in a neighbourhood defined by the **x_width** and **y_width** parameters is computed and if the pixel value differs by more than **kappa_sigma_thres** times the computed sigma, the pixel is flagged as bad. This procedure is iterated until no new bad pixels are detected. While this procedure is very effective at pinpointing bad pixels, it is also computationally intensive, and thus very time consuming. For this reason it is disabled by default.

If the **apply_cuts_switch** is set to YES, each pixel is compared with the validity interval provided by the **low_cut** and **high_cut** parameters and any pixels outside this interval are flagged as bad. This is meant to quickly catch saturated pixels and/or bad values due to electronics. The cuts must be chosen keeping in mind whether the input frame was already bias-subtracted or not, and leave enough allowance for normal noise, in order not to cause a spurious mass rejection of pixels.

The final bad pixel mask will thus be the merged set of all bad pixels detected by any means, or previously marked as such in a user-defined bad pixel frame whose name(s) was (were) set in the **BADPXFRAME** keyword.

11.1.3 Bad pixel map determination

The location of the traps and dead columns on the CCD chip are mapped according to the instrument setting (wave, bin, CCD chip), as determined by examining the template image given. In addition, all pixels above the given saturation threshold are marked as bad. With this information a bad-pixel image is created, that will be used in the following data reduction to mask appropriately the bad pixels.

11.1.4 Fibre order tracing

This data reduction step finds all "traces" in a set of fibre flat-field frames, in which lit fibres do not overlap, typically an even and an odd fibres frame. The **nb_traces** parameter, if set, forces the software to detect exactly this number of traces.

The traces are found using the Hough transform technique. To improve robustness of the wavelength calibration and the extraction in different settings we have set values of **SCAN** keyword with the FITS keywords **DRS_SCAN_MIN** (set to 55,73,73) and **DRS_SCAN_MAX** (set to 1993,1975,1975) to prevent the case in which it could be detected very small portion of fibres in the two corners in the bottom right hand side or in the top right hand side of the detector.

While sensible defaults are provided for them, some of the above parameters can be given on the command line and are passed along unchanged, but it is strongly advised to avoid changing them if their meaning is not clear to the user.

These traces are then correlated with the order positions from the first guess order table, and labelled according to order and fibre number as deduced from the **FIBREMASK** keyword in the fibre flat field frames.

The orders are then fitted with a polynomial of degrees as defined in the **DEFPOL** keyword, assuming constant relative positions of the fibres. The polynomial coefficients are stored in the **COEFFD** descriptor of the output order-fibre table, while the relative position of the fibres are stored in the **FIBREPOS** FITS keyword of the same table.

11.1.5 Background determination

This data reduction step takes as input the order table (e.g. the table raworder.fits created by flames_cal_predict to create a table of positions where the scattered-light background may be computed, in regions not illuminated by any fibre.

The user is advised to recreate the background table using the order-fibre table from the flames_cal_prep_ofpos recipe before starting optimal (standard) extraction. In general, when a background table and an order-fibre table are required for a DRS command, it is considered good practice to freshly create the background table using the same order-fibre table to be subsequently used.

11.1.6 Measure of flat part size of Slit frames

Finds the **FWHM** across y of the illuminated part of a slit flat-field frame near frame center, finds the offset (y shift) of the order centres in this frame with respect to order positions as traced by the order table, and writes these data as keywords **YSHIFT** and **HALFWIDTH** of the input slit flat-field frame.

11.1.7 Slit frames preparation

This command takes a set of calibration slit flat-field frames (at least 2 frames), ordered by Y shift and rescaled in intensity so that each frame intensity is the same as the one of the previous adjacent one where the two overlap, and creates a new, minimal set of slit flat-field frames plus other auxiliary frames (normalization, boundary data) used by subsequent DRS commands. On each input slit flat field frame, the **HALFWIDTH** and **YSHIFT** descriptors should be set, defining respectively the half width of the flat part of the orders and the offset between the order centres in this frame and the order centres as traced by the order-fibre table (see 11.1.6).

Of course, for relative equalisation to be possible, each frame *must* have some overlap with the neighbouring ones in the set.

11.1.8 Fiber frames preparation

Takes as input the same odd/even fibre flat-field frames which were used for order-fibre detection, normalizes them (each fibre is normalised to 1 integrated flux at each x position), and selects only fibres falling within the region illuminated in the slit flat-field frames, to create a new set of odd/even fibre flat-field frames plus other auxiliary frames (normalization, fibre boundaries) used by subsequent DRS commands. Any bad pixels in the input fibre flat-fields are filled with interpolated values, if possible. Each fibre is then correlated with a Gaussian centred at the position traced by the order-fibre table, with a variable offset.

The variable y offset is allowed to span the interval between **-MAXYSHIFT** and **+MAXYSHIFT**.

The half width at half maximum of the Gaussian pseudo-fibres is read from the **GAUSSFIBRESIGMA** keyword, and their total half width is read from the **GAUSSHALFWIDTH** keyword.

The offset corresponding to the maximum correlation is saved in descriptors, to be used as a “zero-point” correction for fibre asymmetry when measuring actual fibre offsets on other frames. For this correction to be effective, subsequent correlations are forced to use the same Gaussian parameters (i. e. **GAUSSFIBRESIGMA** and **GAUSSHALFWIDTH**) which were used here.

11.1.9 Fiber frames normalization

This command performs an optimal or standard (according to the value chosen for the parameter **method**) spectrum extraction on the all-fibre flat-field frame.

- First, if the **BKGFITINLINE** keyword is set to “yes” (as it is by default), a polynomial background is fitted to the positions in the **back_table**, and subtracted from the **all_fibre_ff_frame**. For more details about how this is done and which keywords control it. Setting **BKGFITINLINE** to "no" disables inline background fitting and subtraction.

- Then the y-axis shift of the orders/fibres in the **all_fibre_ff_frame** is found by correlating the frame with a synthetic one composed of Gaussian-shaped pseudo-fibres centred on the positions traced by the order-fibre table, with a variable offset in the direction perpendicular to dispersion. The maximum of said correlation is found, and the corresponding offset is corrected for the "zero-point" shift. This y offsets are saved as a descriptor called **YSHIFT** in the science frame. The correlation step can be disabled by setting the **max_shift** parameter to zero or, equivalently, setting to zero the **MAXYSHIFT** keyword and leaving the **max_shift** parameter blank. In this case, the values already contained in the **YSHIFT** descriptor are used.

- The normalised fibre flat-field frames are then shifted to bring them to coincidence with the above offset and are multiplied by the slit flat-field frames to approximate fibre flat-field frames exactly matching the all-fibre flat-field frame. These frames are used to perform an optimal or standard extraction on the science frame, including a deconvolution of adjacent fibres (the deconvolution coefficients are computed directly from the odd/even fibre flat-fields).

- The extracted "spectra" are saved to be later used as relative normalisation factors between fibres in the subsequent extraction of science frames.

11.1.10 Fiber frames extraction

This data reduction step performs an optimal or standard normal spectrum extraction. Are supported three modes: "normal" (**extract**=opt/sta), the one suggested to the user; "fast" (**extract**=fop/fst), which uses input slit flat fields to clean the fibre PSFs but skips the steps of Y-shift determination and compensation (see below), a bit faster than "normal"; "quick" (**extract**=qop/qst) which does not uses the slit flat fields and does not correct for fibre shifts, the fast and less accurate mode.

- Background determination. First, if the **BKGFITINLINE** keyword is set to "yes" (as it is by default),a polynomial background is fitted to the positions in the **back_table**, and subtracted from the **science_frame**. For more details about how this is done and which keywords control it. Setting **BKGFITINLINE** to "no" disables inline background fitting and subtraction.

- Y-shift determination. Then the y-axis shift of the orders/fibres in the science_frame is measured correlating the frame with a synthetic one composed of Gaussian-shaped pseudo-fibres centred on the positions traced by the order-fibre table, with a variable offset in the direction perpendicular to dispersion. The maximum of this correlation function is found and the corresponding offset is corrected for the "zero-point" shift. For robustness, this step comprises a preliminar correlation function shape determination and rough search for its maximum. This feature can be deactivated by the keyword(s) **DRS_COR_MAX_FND**, **DRS_COR_DEF_RNG**, **DRS_COR_DEF_PNT**. This y offset is saved as a FITS keyword called **YSHIFT** in the science frame. The correlation step can be disabled by setting the **max_shift** parameter to zero or, equivalently, setting to zero the **MAXYSHIFT** keyword and leaving the **max_shift** parameter blank. In this case, the values already contained in the **YSHIFT** FITS keyword are used.

- Y-shift correction. The normalised fibre flat-field frames are then shifted to bring them to coincidence with the above offset and multiplied by the slit flat-field frames, to approximate fibre flat-field frames exactly matching the science frame.
- Extraction. The frames determined as explained above are then used to perform an optimal extraction on the science frame, including a deconvolution of adjacent fibres (the deconvolution coefficients are computed directly from the odd/even fibre flat-fields).

Notes: Should the correlation step fail to detect a reasonable y shift, or if speed is of utmost importance, it is advisable to use standard extraction, disabling the correlation and reducing the integration window to the core of the fibres only, rather than optimal extraction, to minimise fibre to fibre contamination, at the price of losing a considerable fraction of the signal.

In cases of well-behaved set of frames, however, optimal extraction gives the best signal/noise ratio, especially for faint objects. The calculated spectra are finally written on the disk on files in MIDAS format, one set of files for each illuminated fibre.

The standard extraction includes the deconvolution of adjacent fibres, be them neighbouring fibres of the same order or the first and last fibre of adjacent orders. The deconvolution coefficients are computed directly from the shifted fibre flat-field frames.

Since in the case of fast optimal extraction no attempt at all is made to compensate for any y offset of the science frame with respect to the normalised fibre flat-field frames, this method should be selected with some caution.

In cases in which the y offset determination is a problem, the fast extraction should be used instead, limiting the integration window to the cores of the fibres to minimize ill effects at the cost of losing a considerable fraction of the signal.

11.1.11 Fiber frames merging

This command first rebin the extracted spectrum, its variance and its bad-pixel mask.

Then it performs optimal merging of the orders, computing every pixel in the merged spectrum as a weighted average of all good pixels available covering its wavelength bin, with weights equal to the inverse of the respective variances.

Important information is contained in the binary table extensions. The most important information about the observed object (object id, its magnitude, right ascension and declination, fiber id) is displaied together with the reduced spectra using the command plot/uves (see section **??** at page **??**).

11.1.12 Interorder background subtraction

Two different methods are used to subtract the scattered light background:

- Spline method. The input image is sampled at half-integer order locations at **backsub.npoints** equally spaced sample points. According to the user defined value of the parameter **backsub.mmethod**, the median or the minimum values computed in a subwindow of height (2 * **backsub.radiusy**/biny_size + 1). The window width is given by the distance between the sample positions.

After the spline interpolation, the interorder background image is filtered using an average filter with radius (**backsub.smoothx, backsub.smoothy**).

The spline degree is set by using **backsub.sdegree**. Currently, only splines of degree 1 are supported (i.e. linear interpolation). If the **backsub.mmethod** parameter is set to 'no', no background subtraction is done.

- Polynomial method. The polynomial method is used in the order definition recipe because the order locations, required for the spline method, are not known at this initial stage.

  A low degree 2d polynomial is fitted to a subset of the image pixels and outlier points (such as the orders themselves) that have large positive residuals are continuously rejected (one-sided kappa-sigma clipping).

  The input image is sampled on a regular grid with mask size (**image_width / backsub.npoints , image_height / backsub.npoints)**.

  After the initial order line detection, the interorder background is sampled (at locations separated by **image_width / backsub.npoints**) between the order lines.

11.1.13 Hough transform

The Hough transform is computed by sampling the input image at every column seperated by **samplewidth**.

Each echelle order maps to a peak in the Hough image. After detecting a peak, the peak itself and the area around it are deleted to avoid redetecting the same feature.

The accurate peak locations are calculated as the centroid of the area around the local maximum in the Hough space.

The number of order lines to detect is specified by the parameter **norders**. If this parameter is set to zero, the function will detect lines until the intensity of the next candidate drops to below a fraction **pthres** of the dimmest line.

An important parameter for the peak removal to work is the (approximate) interorder spacing. This parameter is estimated as the raw image height divided by the predicted number of orders (**norders**). In automatic mode it is estimated as the first minimum of the auto-correlation function along the column in the Hough image which contains the global maximum. This fully automatic way of detecting the orders assumes only that the interorder spacing does not vary too much as function of order number.

Possible order line slopes range from **minslope** to **maxslope** The resolution of the slope-axis in Hough space is defined by **sloperes**.

11.1.14 Order tracing

The parameter **use_guess_tab** value, defaulted to 1, allow the user to benefit of the information contained in the guess order table in a different way:

- 0: No usage.

- 1: Use the guess order table to set the lower/upper Y raws where order are searched.
- 2: The order table try to fully match the guess one.

The order tracing is performed as follows:

- First, all orders are traced in both directions starting from the center which is inferred from the solution of the Hough detection (if **use_guess_tab** is set to 0), or from the "guess" order table (if **use_guess_tab** is set to 1 or 2).

  A Gaussian is fit to the order line at x-positions seperated by the parameter **tracestep**. The trace stops if the intensity of the order line drops below the threshold defined by the **minthresh** value in an x-range determined by the parameter **maxgap**.
- Then each order is fitted with a straight line, and the entire order is rejected if the RMS is large compared with the average RMS.
- A global polynomial of automatic degree is fitted to all orders, and individual points are rejected using kappa-sigma clipping. Alternatively, the user can define polynomial degrees using the parameters (**defpol1, defpol2**).

11.1.15 Line Search

A number of emission lines defined by the range **minlines-maxlines** (both inclusive) is searched for in the extracted arc lamp spectrum. This is achieved by adjusting the detection threshold level until an appropriate number of lines is detected.

A 5-parameters Gaussian fit, including the continuum slope, is made if the line peak is above the treshold with respect to the local interorder background level, which is defined as the median of a window of width (2 * **range** + 1) centered on the current position. Finally, doublets (defined as lines with positions within 2.0 pixels) are removed from the set of detected lines.

11.1.16 Wavelength calibration first solution determination

An initial dispersion relation is obtained by fitting the relation

$$\lambda * m = f(x, m),$$

to a guess line table containing associations from (x, m) = (pixel, order) to wavelengths. Here f is a 2d polynomial.

A systematic x-shift up to **shiftmax** pixels is recovered by finding the maximum position of the cross-correlation function and applying this shift to the inital dispersion solution. The resolution of the cross-correlation function is defined by **shiftstep**. The parameter **shifttoler** defines the tolerance in pixels for the line match. The default polynomial **degree** is 5.

11.1.17 Line identification

The wavelength calibration starts from a first guess dispersion solution.

Lines are iteratively identified and a dispersion solution is fitted, until no new identifications can be made. After the first convergence, all identifications are cleared (to remove possible false identifications), and the loop repeats, but this time ignoring lines with residuals worse than **tolerance** and lines with residuals worse than **kappa** sigma. If set to a negative value, the polynomial degree is automatically adjusted based on the line residuals.

Identifications are made based on a match between the detected line predicted wavelength, λ_{com} and a catalogue wavelength, λ_{cat}. An identification is made if

- The nearest catalogue wavelength is within two linewidths of the predicted wavelength: $|\lambda\_cat - \lambda\_com| < 2 * \sigma$, where σ is the detected line width,
- The distance to the 2nd nearest neighbours (in the spectrum as well as in the catalogue) is much larger than the residual of the match $|\lambda\_cat - \lambda\_nn| * ALPHA > |\lambda\_cat - \lambda\_com|$, (where **ALPHA** is a "safety parameter" less than one.)
- The nearest neighbour (in the spectrum and in the catalogue) is farther away than the average tolerance distance, which measures the precision of the identifications: $\mathbf{tolerance} < \mathbf{ALPHA} * |\boldsymbol{\lambda}\mathbf{\_cat} - \boldsymbol{\lambda}\mathbf{\_nn}|$. Refer to the source code for the exact definition of the $\mathbf{tolerance}$.

The purpose of the first criterion is to make the correct identifications. The purpose of the latter two criterions is to avoid making incorrect identifications.

11.2 Recipes

In the following sections we are going to describe the recipes provided by the UVES pipeline. Recipe common parameters are: **debug**, to activate the debug mode which saves intermediate results to disk and **plotter** to activate the plotting facility (gnuplot). Additionally each recipe can be customized by modifying specific data reduction parameters which are described in the following section or in the referred algorithm description.

11.2.1 flames_cal_predict

This recipe expects as input a fibre formatcheck frame which is obtained by illuminating the calibration fibre with a ThAr lamp, and a reference ThAr lines table. Optional input is a reference formatcheck frame.

This recipe run the UVES physical model as for ECHELLE data. For details we remind the user to [1]. The only difference is that in case of FIBER mode data, to take into account of the offset between the calibration fiber and the central position of the order, X and Y offsets are set depending on the given instrument setting (wavelength, plate number).

11.2.2 flames_cal_orderpos

This recipe measures the echelle calibration fibre order trace positions as for echelle data. For details we remind the user to [1].

11.2.3 uves_cal_mbias

This recipe generates a master bias frame from a set of biases as for echelle data. For details we remind the user to [1].

11.2.4 uves_cal_mdark

This recipe generates a master dark frame from a set of darks (darks are not frequently taken for UVES). For details we remind the user to [1].

11.2.5 flames_cal_mkmaster

This recipe generates a master flat frame from a set of slit flats.

To have full coverage of all 9 fibres and wide enough windows to determine the inter order background are taken three sets of three slit flats each, each set taken at different Y positions. From each set it is computed a master flat field as for echelle mode data. For details we remind the user to [1].

11.2.6 flames_cal_prep_sff_ofpos

This recipe prepares slit flat-field and fibre flat-field frames for inclusion in the calibration database. The command reduces first the lower-chip data and next the upper-chip data. The slit FF frames are normalised to be equal where they overlap, then a minimal set of frames covering the largest y interval is produced.

This command

- Creates suitable bad-pixel masks,
- Frames are prepared (to associate variance and bad-pixel frames to existing frames),
- All fibre and order positions are located,
- Suitable position for background-light extraction are located,
- input slit flat-fields are prepared,
- input fibre flat-fields are prepared,
- fibre-to-fibre normalizations are computed.
- Last, all output products are classified.

To improve robustness this data reduction step checks that for each detected order all the fibres are traced and if not a fibre tracing is forced. Moreover the step OFPOS, to prevent problems in the wavelength calibrations, excludes from the order tracing the bottom right and upper left hand side corners. This is done setting appropriate values of keywords **DRS_SCAN_MIN** and **DRS_SCAN_MAX**, and using an appropriate value for the parameter **DRS_P8_OFPOS_S1**. Such values are wavelength dependent.

11.2.7 flames_cal_wavecal

This recipe performs the wavelength calibration using previously determined solutions for the first guess line table and the order table. This recipe is a modification of the **uves_cal_wavecal** recipe to reduce UVES echelle data, in the sense that instead of determine the solution corresponding to tree order traces, one at the center and the other two at 15 pixels up and down, now are determined the solutions correspoinding to each of the fiber traces as indicated by the **FIBREMASK**. FITS keyword. For details we remind the user to [1].

11.2.8 flames_obs_reduce

This recipe performs the actual spectrum extraction, either optimal or standard. The normalized odd/even fibre flat-field frames output of previous commands are compared to the science frame, any y-axis shift between them is computed and compensated by shifting the odd/even fibre flat-fields; an optimal/standard extraction is performed on the science frame, and the resulting spectra are created, one set of files for each illuminated fibre. Next, using the line table created by **flames_cal_wavecal**, the extracted spectra are rebinned and merged. Finally, the products are classified. This commands supports different extraction modes. The normal extraction mode (the one just described, using optimal or standard extraction, setting the parameter **extract** to "opt" or "sta") is the one we suggest to the user. Fast extraction mode (**extract** set to "fop" of "fst") is faster as skip the correlation step (but as such it may have also lower quality). Quick extraction mode, which does not use at all the slit flat fields in the data reduction chain, and does not do any correlation step, was implemented in answer to a Paranal Operation request, and if used, requires that the complete data reduction (recipes **flames_cal_prep_sff_ofpos** is performed in a coherent way (using quick data reduction mode).

In case the correlation step fails (usually due to a large Y shift between the observation and the calibrations) we suggest the user to do the following. The shape of the correlation function is contained in tables cor_shape_x.fits (x=l or x=u respectively for EEV or MIT chips). From this table one can get the Y offset at which the correlation function has a maximum.

Next repeat the extraction appropriately setting the correlation function parameters (through parameter `extract`). In this case usually **drs_cor_def_rng** can be decreased to 2 or even 1.

For example to do optimal extraction with correlation (OPT) function search (Y) in the range [-2,2], using 2*25+1 pixels and having measured a -4.5 pixel shift, one may give the command:

```
esorex flames_obs_reduce --cor_pnt=25 --cor_rng=2 --cor_off=4.5 flames_obs_reduce.sof
```

11.2.9 flames_obs_redchain

This recipe runs the full UVES reduction chain for the blue and/or red arm. It runs in a cascade the following recipes:

- uves_cal_mbias (if no master bias is provided)
- uves_cal_mdark (if darks are provided)
- flames_cal_predict (if no guess order and line tables are provided)
- flames_cal_orderpos (if no order table is provided)
- flames_cal_mflat (if no master slit flat is provided)
- flames_cal_prep_sff_ofpos (if no slitff* and fibreff* are provided)
- flames_cal_wavecal (if no dispersion solution line table is provided)
- flames_obs_scired (unless the option **scired** is set to FALSE)

A Installation

This chapter gives instructions on how to obtain, build and install the UVES pipeline. Even if this chapter is kept as up-to-date 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 UVES pipeline distribution kit. These release-specific instructions can be found in the file `README` located in the top-level directory of the unpacked UVES pipeline source tree. The supported platforms are listed in Section A.1. It is recommended reading through Section A.3 before starting the installation.

A bundled version of the UVES pipeline with all the required tools and an installer script is available from `www.eso.org/pipelines`.

A.1 Supported platforms

The UVES pipeline has been verified to install and execute correctly with EsoRex on the VLT target platforms:

- Linux (glibc 2.1),
- Sun Solaris 5.8,

and on

- Mac Darwin 8.9.0,

using the GNU C compiler (version 3.2 or newer). Correct execution using Gasgano has been verified on

- Scientific Linux 4.0,
- Linux Fedora core 6,
- Mac Darwin 8.9.0.

A.2 Requirements

To compile and install the UVES pipeline one needs:

- the GNU C compiler (version 3.2 or later),
- the GNU `gzip` data compression program,
- a version of the `tar` file-archiving program and
- the GNU `make` utility.

For Gasgano support one needs in addition

- the Java Development Kit (version 1.5)

A.3 Building the UVES pipeline

The UVES pipeline distribution kit contains:

uves-manual-5.0.pdf	The UVES pipeline manual
install_pipeline	Install script
qfits-6.2.0.tar.gz	QFITS 6.2.0
cfitsio-2510.tar.gz	CFITSIO 2510
cpl-4.1.0.tar.gz	CPL 4.1.0
esorex-3.6.8.tar.gz	esorex 3.6.8
gasgano-2.2.7.tar.gz	GASGANO 2.2.7
uves-4.2.2.tar.gz	UVES 4.2.2
uves-calib-4.2.2.tar.gz	UVES static calibration files 4.2.2

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the UVES pipeline 4.2.2 package. It can be any directory of your choice but not:

   ```
   $HOME/gasgano
   $HOME/.esorex
   ```

2. Download from the ESO ftp server, `www.eso.org/pipelines`, the latest release of the UVES pipeline distribution.

3. Verify the checksum value of the tar file with the cksum command. **cksum uves-kit-4.2.2.tar.gz**

4. Unpack using the following commands:
   gunzip uves-kit-4.2.2.tar.gz tar -xvf uves-kit-4.2.2.tar

   Note that the size of the installed software (including *Gasgano*) together with the static calibration data is about 27Mb.

5. Set the environment variable JAVA_HOME to the directory where you have the JDK 1.5 installed. If this value is not set, the installation script will try to guess it, but if no JDK is found, the gasgano distribution will not be installed; QFits, CPL, EsoRex and the pipeline will be installed anyway.

6. Install: after moving to the top installation directory,
   cd uves-kit-4.2.2

   it is possible to perform a simple installation using the available installer script (*recommended*):
   ./install_pipeline

7. Check the installation log: probably this will suggest you to set the environment variable CPLDIR and to extend your PATH.

By default the script will install the UVES recipes, *Gasgano*, *EsoRex*, all the necessary libraries, and the static calibration tables, into a directory tree rooted at `$HOME`. A different path may be specified as soon as the script is run.

The only exception to all this is the *Gasgano* tool, that, if you have the required proper installation of the JDK (version 1.5), will always be installed under the directory `$HOME/gasgano`. Note that the installer will move an existing `$HOME/gasgano` directory to `$HOME/gasgano.old` before the new *Gasgano* version is installed.

Important: the installation script would ensure that any existing *Gasgano* and *EsoRex* setup would be inherited into the newly installed configuration files (avoiding in this way any conflict with other installed instrument pipelines).

B Full list of changes

```
List of differences between UVES-CPL pipeline releases 4.2.2 and 3.4.5

Changes in 4.2.2
        DFS05346 Problem wavecal/merge: absorption lines may appear as doublets
        in merged spectra. Solved by decreasing wavecal poly degree default
        to 4.

Changes in 4.2.1
        Added uves_cal_ronbias
        DFS05302 Added PRO.REC1 keys in uves_cal_prep_sff_ofpos products
        DFS05106 Now header of slit_ff_products is the one of the 1st frame of the
        DFS05102 Implemented flames_utl_decubify
Changes in 4.2.0
        DFS05140: changed LF.DO.CATG in DO.CATG for CD_ALIGN_RED to uniform to CD_
        DFS05139: Updated gasgano rule files to support classification of
        input raw data for uves_cal_lingain recipe
        Added products of uves_cal_lingain to oca rules.
Changes in 4.1.0
        Removed uves_cal_ccdtest from oca file
        Added flames_cal_mkmaster in UVES.prefs
        Disables uves_cal_ronbias
        Updated cpl.version and esorex.version dependency IDs
Changes in 4.0.0
        As 3.5.7 but updated release ID for PSO release (as includes fiber mode)
Changes in 3.5.7
        Added parameter clean_traps (shared in uves_obs_scired and
        uves_cal_mbias) to control the detector's trap corrections
Changes in 3.5.6
        DFS05042: Fixed wrong INFO in FIB_FF_*_INFO_TAB
        Fixed DFS04897: Failure to trace first fibre of FFLAT
        added parameter save_flat_size to flames_cal_prep_sff_ofpos to control
        SFLAT flat size measure
        Fixed DFS05026: Missing PIPEFILE in fibreff cube products.

Changes in 3.5.5
        Made uves_cal_lingain products arm dependent (note that in case of RED da
        DFS03207: Poor fibre identification in Fibre Flats. This problem has been
        improving robustness of the function to measure the MSFLAT FWHM and YSHIF
        In flames_cal_prep_sff_ofpos added parameter to control on which chip is p
        DFS05006: Added parameter to allow clean of trap columns in science
        frames (to improve 2D data reduction quality)
        Fixed bug loading pipe generated INSTR_RESPONSE
        Demoted assure on check_table_invalid_rows to make
```

```
        flames_cal_prep_sff_ofpos more robust on MIDAS generated FIB_GUE_ORD_TAB
Changes in 3.5.4
        Changed order of LDFLAGS in uves/Makefile.am and flames/Makefile.am
        to make sure that libuves.so and libflames.so always find the same
        version of libqfits as each other.
Changes in 3.5.3
        More robust SFLAT size-offset determination
        Support of BLUE 625 KHz frames format with image estension.
        Improved uves_common.prg including DETMON_QC dictionary call
        Simplified uves_cal_lingain
        Cleaned some compilation warnings
Changes in 3.5.2
        Updated uves_cal_lingain to support BIAS_x and SCREEN_FLAT_x, x=BLUE,RED
        Aadded uves_cal_lingain.rrd,rri,prg
Changes in 3.5.1
        Implemented new function to compute YSHIFT and HALFWIDTH on mflats
        to solve DFS04815

Changes in 3.5.0
        Added HIERARCH ESO FIB keys in flames science packed merged frames (DFS04
        DFS04881: allow usage of parameter process_chip to process each chip sepa
        Temporaly suppressed velocity correlation correction in SimCal
        Added test_flames_msffsz
Changes in 3.4.9
        Fixed DFS04829: (zero merged spectra in packed image)

        Added uves_cal_lingain from detmon project

        Lamp on frames (FLAT frames) should be tagged as ON_RAW

        Lamp off frames (BIAS frames) should be tagged as OFF_RAW

        New FITS format UVES data (after 1st April 2004) should be processed
        using --exts=1

Changes in 3.4.8
        Now the stability check computes a clean median as in MIDAS
Changes in 3.4.7
        Removed flames recipes
        Added HIERARCH ESO info on debug products
        Fixed some problems with flames physical model and reduced to 1 the numbe

Changes in 3.4.6
        Added flames recipes
        Fixed PRO.CATG of fxb_l_rawpack XB_SCI_RAW_REDL
```

```
and added ERR_MWXB_SCI_RAW_x and ERR_MWXB_SCI_x as asked by DFS04684
```

C Abbreviations and acronyms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CalibDB	Calibration Database
CPL	Common Pipeline Library
DFO	Data Flow Operations department
DFS	Data Flow System department
DMD	Data Management and Operations Division
DRS	Data Reduction System
ESO	European Southern Observatory
EsoRex	ESO-Recipe Execution tool
FITS	Flexible Image Transport System
FOV	Field Of View
FPN	Fixed Patter Noise
GUI	Graphical User Interface
OB	Observation Block
PSO	Paranal Science Operations
QC	Quality Control
RON	Read Out Noise
UVES	Ultraviolet Visual Echelle Spectrograph
SOF	Set Of Frames
UT	Unit Telescope
VLT	Very Large Telescope

Index