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

UVES Pipeline User Manual

VLT-MAN-ESO-19500-2965

Issue 12.0

Date 2008-11-28

Prepared:	Jonas Møller Larsen and Andrea Modigliani	2008-11-27	
	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
7.5	01/04/2007	All	Internal release to PSO/DFO
9.0	15/06/2007	All	First release after conversion to CPL
10.0	18/12/2007		Update to 3.4.5
11.0	18/04/2008		Update to 3.9.0
12.0	28/11/2008		Update to 4.3.0

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

The 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 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 UVES data reduction sequence with the UVES pipeline.

This manual is a complete description of the data reduction recipes implemented by the the CPL based UVES pipeline, reflecting the status of the UVES pipeline as of Nov 28, 2008 (version 4.3.0). Release 4.3.0 supports the reduction of UVES frames obtained in the UVES standard modes. [1]

1.2 Acknowledgements

The UVES pipeline has been initially developed by Sebastian Wolf and Olivier Boitquin under the supervision and responsibility of Pascal Ballester. It has been designed and implemented in the MIDAS environment as a particular application of the ECHELLE context developed by Pascal Ballester. Then it has been put into operations, extended and supported by Andrea Modigliani for more than seven years of operations. He added quality control procedures, the possibility to extract targets keeping the full spatial information, solved more than 120 problem reports and improved the quality of calibrations and extraction. Later he also added the support of the data reduction of data coming from the FLAMES facility, integrating in the MIDAS based UVES pipeline the data reduction software developed by Giacomo Mulas, Ignazio Porceddu and Francesco Damiani. Then, to decrease maintenance costs, the pipeline has been ported to CPL. Andrea Modigliani converted the uves_cal_predict recipe, provided routines and algorithms to generate quality control parameters, contributed to the overall testing and wrote most of this manual. Jonas Møller Larsen converted the remaining recipes and implemented the new optimal extraction algorithms.

The pipeline benefits also from several suggestions and improvements proposed by the UVES Instrument Operations Team members and some users. We would like here to thank explicitly Andreas Kaufer who provided several suggestions at the start of the UVES operations and proposed quality control parameters, Alain Smette, who provided input to improve the optimal extraction, Michael Murphy, who provided improvents to the wavelength calibration and the new ThAr reference catalog. We thank also all the people who patiently and carefully controlled results of this new release, in particular of the science reduction: Cedric Ledoux, Valentina D'Odorico, Emmanuel Jehin, Reinhard Hanuschik, John Pritchard, Huges Sana.

1.3 Scope

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

[1] While the pipeline will generally cope with data acquired in non-standard modes ESO offers NO quality control over such data, and therefore it is responsibility of the user to verify the scientific efficacy of any calibration and/or science products produced by the pipeline for non-standard modes

Updated versions of the present document may be found on [1]. For general information about the current instrument pipeline 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]. Description of the instrument are in [8,9,10]. The UVES instrument user manual is in [10]. The 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 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 http://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/uves
[9] *Successful Commissioning of UVES at Kueyen* The Messenger, **99**, 1, 2000.
[10] VLT UVES User Manual VLT-MAN-ESO-13200-1825
www.eso.org/instruments/uves/doc
[11] VLT UVES Calibration Plan VLT-PLA-ESO-13200-1567
www.eso.org/instruments/uves/doc
http://www.eso.org/instruments/uves/doc/
[12] UVES SV home page
www.eso.org/science/vltsv/ut2sv/fors2_uves.html
[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] The UVES Data Reduction Pipeline The Messenger, **101**, 31, 2000.
[17] Horne: An optimal extraction algorithm for CCD spectroscopy PASP, **98**,609, 1986.
[18] Marsh: The extraction of highly distorted spectra PASP, **101**,1032, 1989.
[19] Mukai: Optimal extraction of cross-dispersed spectra PASP, **102**, 183, 1990.
[20] Murphy et al.: Selection of ThAr lines for wavelength calibration of echelle spectra and implications for variations in the fine-structure constant MNRAS, **378**, 221, 2007

2 Overview

In collaboration with instrument consortia, the Data Flow Systems Department (DFS) 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 88). 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 UVES instrument and the different types of 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 advise 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 UVES pipeline recipes is described, and in Appendix B a list of used abbreviations and acronyms is given.

3 What's new in pipeline release 4.3.0

The following major changes has been realised on the UVES pipeline (with respect to the previous CPL based pipeline):

- Added support for reduction of FIBER mode data.
- Improved data reduction quality of red 860 data (effects due to fringes can be improved setting **reduce.ffmethod** to the 'pixel').
- Added parameters **reduce.merge_delt1** and **reduce.merge_delt2** to uves_obs_scired to control (as in MIDAS based pipeline) the amount of overlapping spectrum considered in merging orders.

4 UVES Instrument Description

UVES has been developed by ESO.

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 UVES instrument is given. A more complete documentation can be found in the UVES User Manual [10].

4.1 Instrument overview

UVES is a two-arms, cross-dispersed echelle spectrograph covering the wavelength range 300 - 500 nm (blue) and 420 - 1100 nm (red) with the possibility to use dichroics (see Figures 4.1.1 and 4.1.2). The spectral resolution for a 1 arcsec slit is about 40,000. The maximum resolution that can be obtained with still adequate sampling, using a narrow slit, is about 110,000 in the red and 80,000 in the blue. For a detailed description of the instruments refer to e.g. [10]. UVES has also a fibre link to FLAMES, the Fibre Large Array Multi-Element Spectrograph, the multi-object, intermediate and high resolution spectrograph mounted at the Nasmith A platform of UT2 of the VLT. The fibre link to the UVES red arm is fed by eight fibres with a nominal resolution power of R=47000.

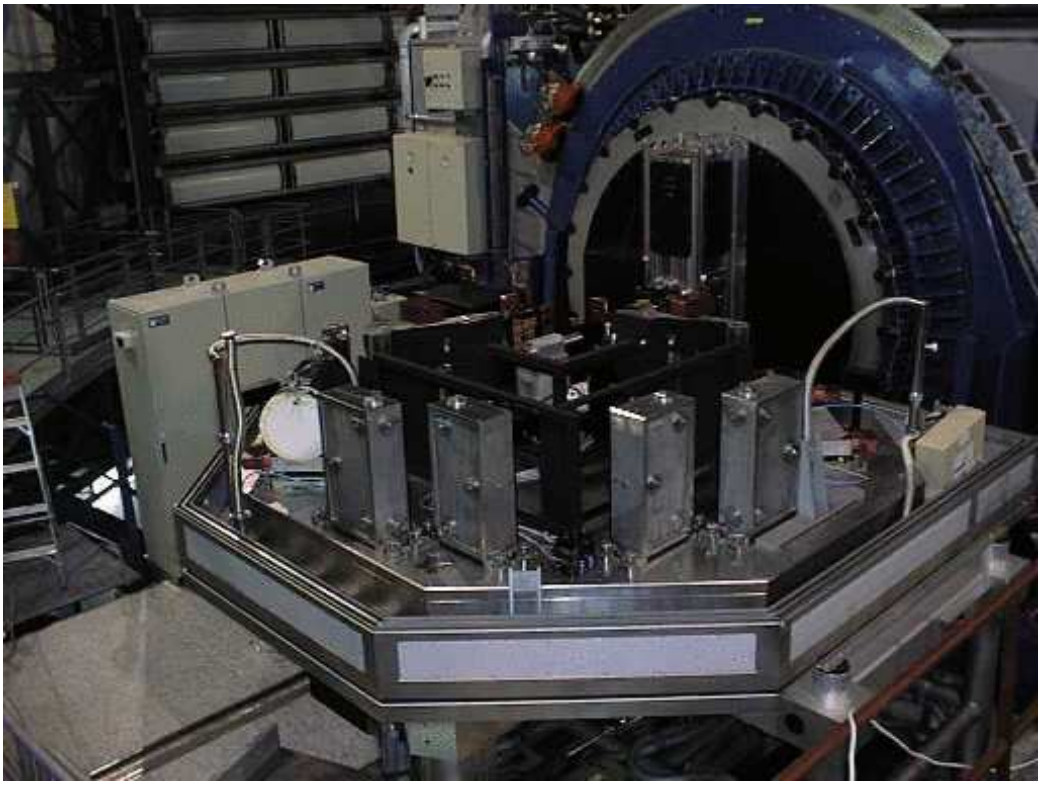


Figure 4.1.1: A photo of UVES mounted at the Nasmyth B focus of Kuyen (VLT-UT2).

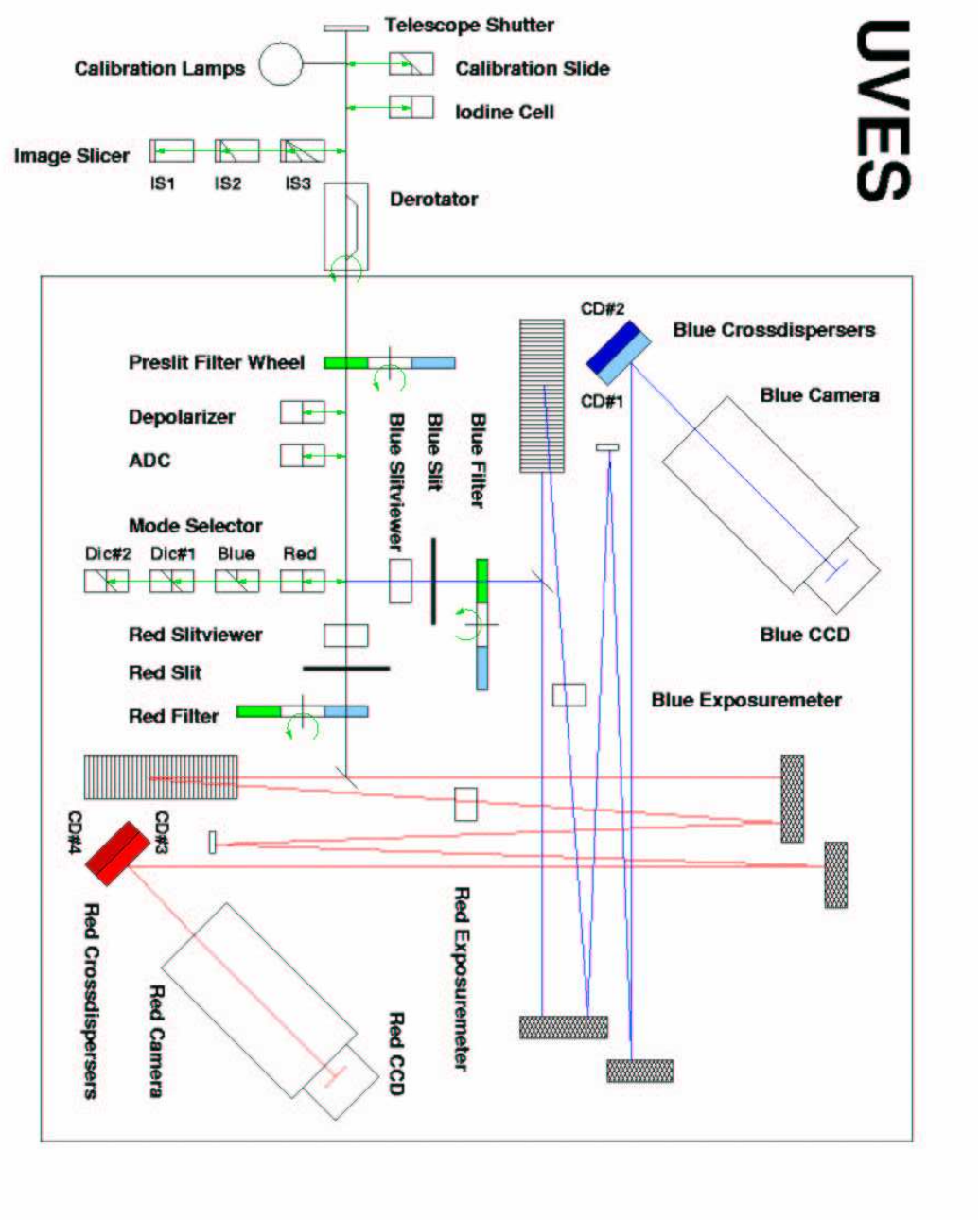


Figure 4.1.2: Right panel: schematic overview of the UVES spectrograph.

5 Quick start

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

5.1 UVES pipeline recipes

The current UVES pipeline is based on a set of 8 stand-alone recipes involved in the data reduction cascade:

uves_cal_mbias creates a master bias frame.

uves_cal_mdark creates a master dark frame.

uves_cal_predict implements the UVES physical model.

uves_cal_orderpos defines echelle order positions.

uves_cal_mflat creates a master flat field frame.

uves_cal_wavecal performs the wavelength calibration.

uves_cal_response determines the response function and quantum efficiency.

uves_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 **uves_obs_redchain** recipe.

Additionally, the pipeline provides 3 recipes:

uves_cal_mkmaster creates the master bias/dark/flat frames. This recipe exists mostly for backwards compatibility. The user is advised to use **uves_cal_mbias**, **uves_cal_mdark** and **uves_cal_mflat** for the reduction of bias/dark/flat frames.

uves_cal_tflat reduces telluric lamp flat field frames (see 7.6). It is used operationally to monitor the instrument performance.

uves_cal_cd_align is used operationally to measure the reproducibility of the cross disperser position.

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

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. 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.

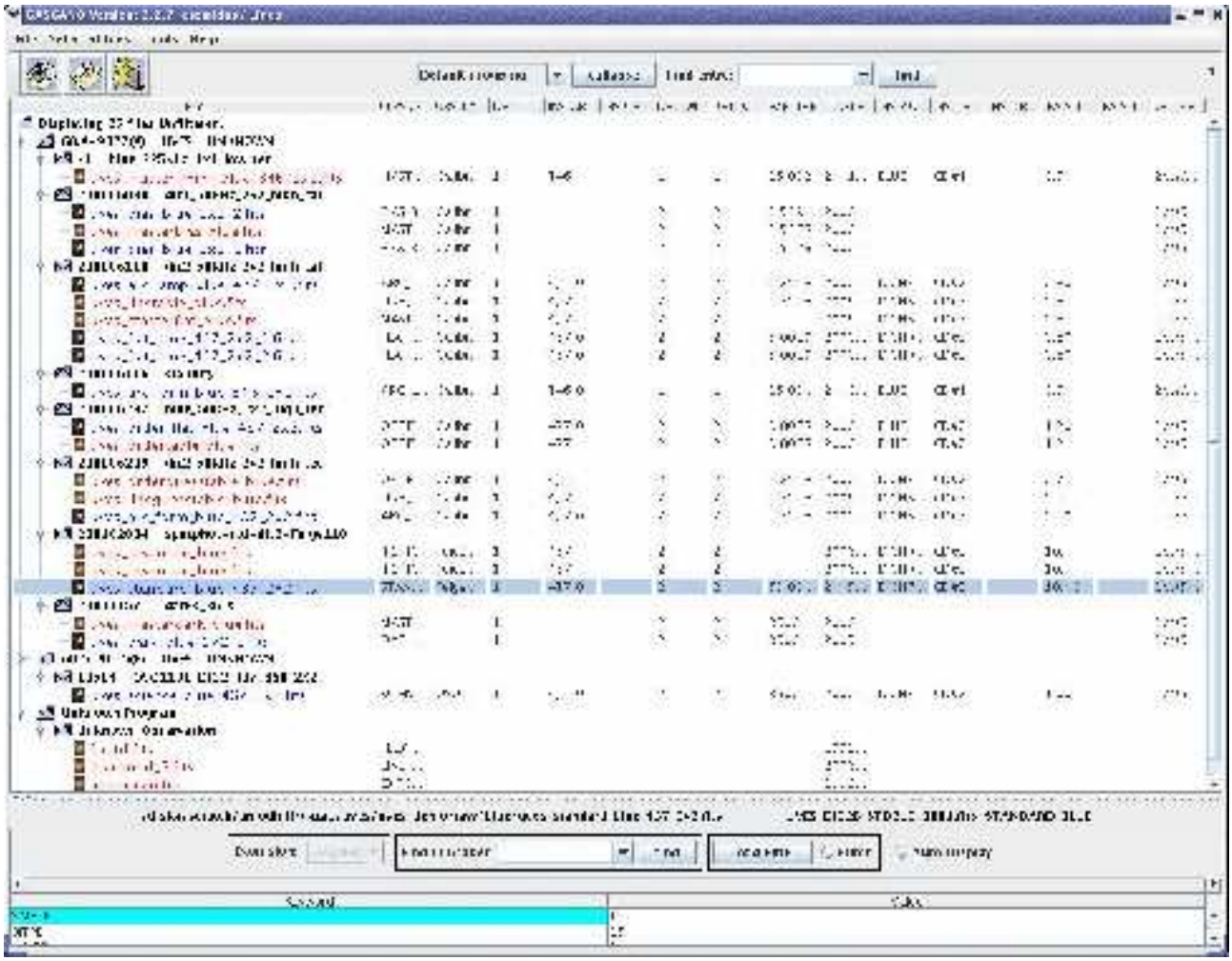


Figure 5.2.1: *The Gasgano main window.*

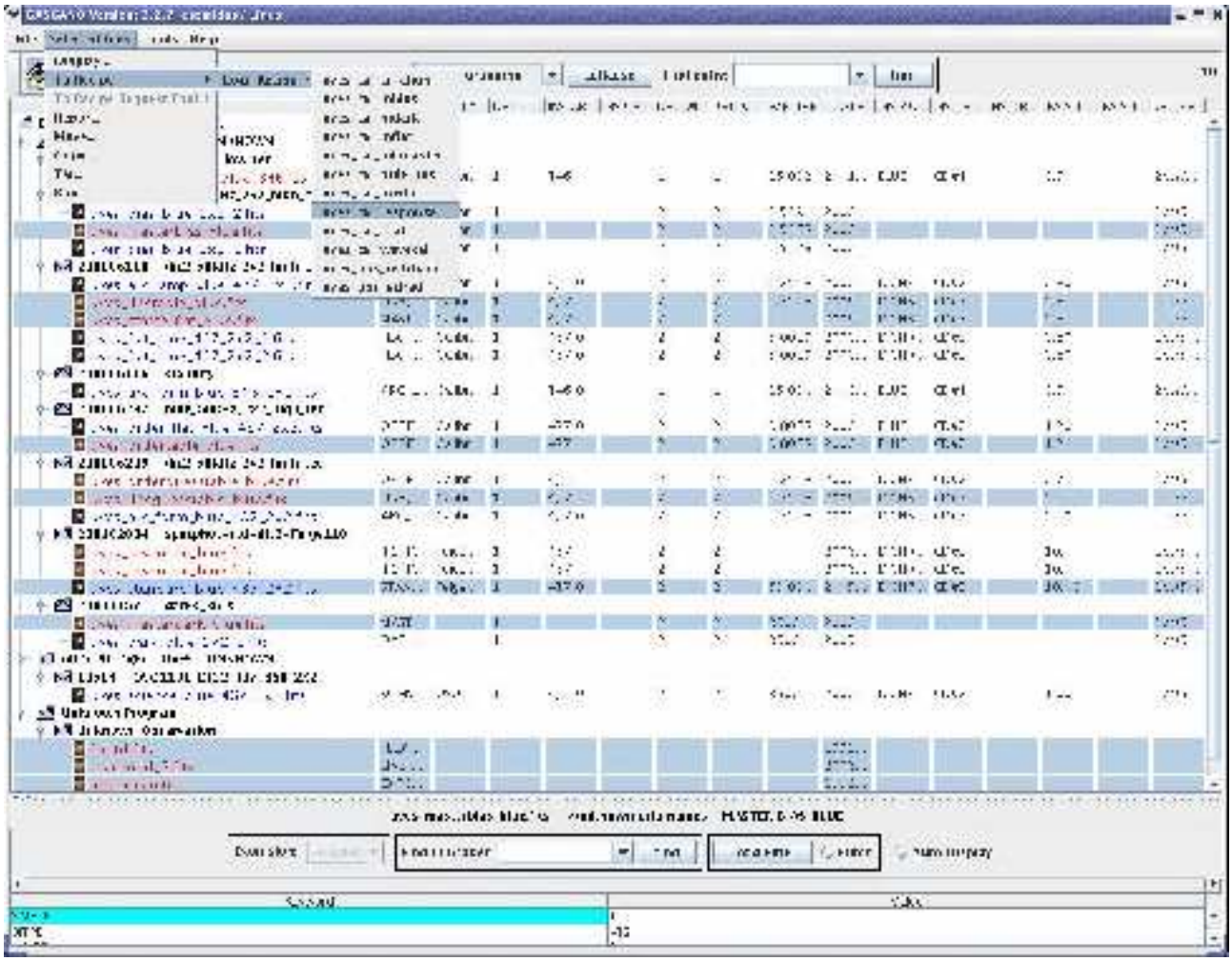


Figure 5.2.2: *Selecting files to be processed by a UVES pipeline recipe.*

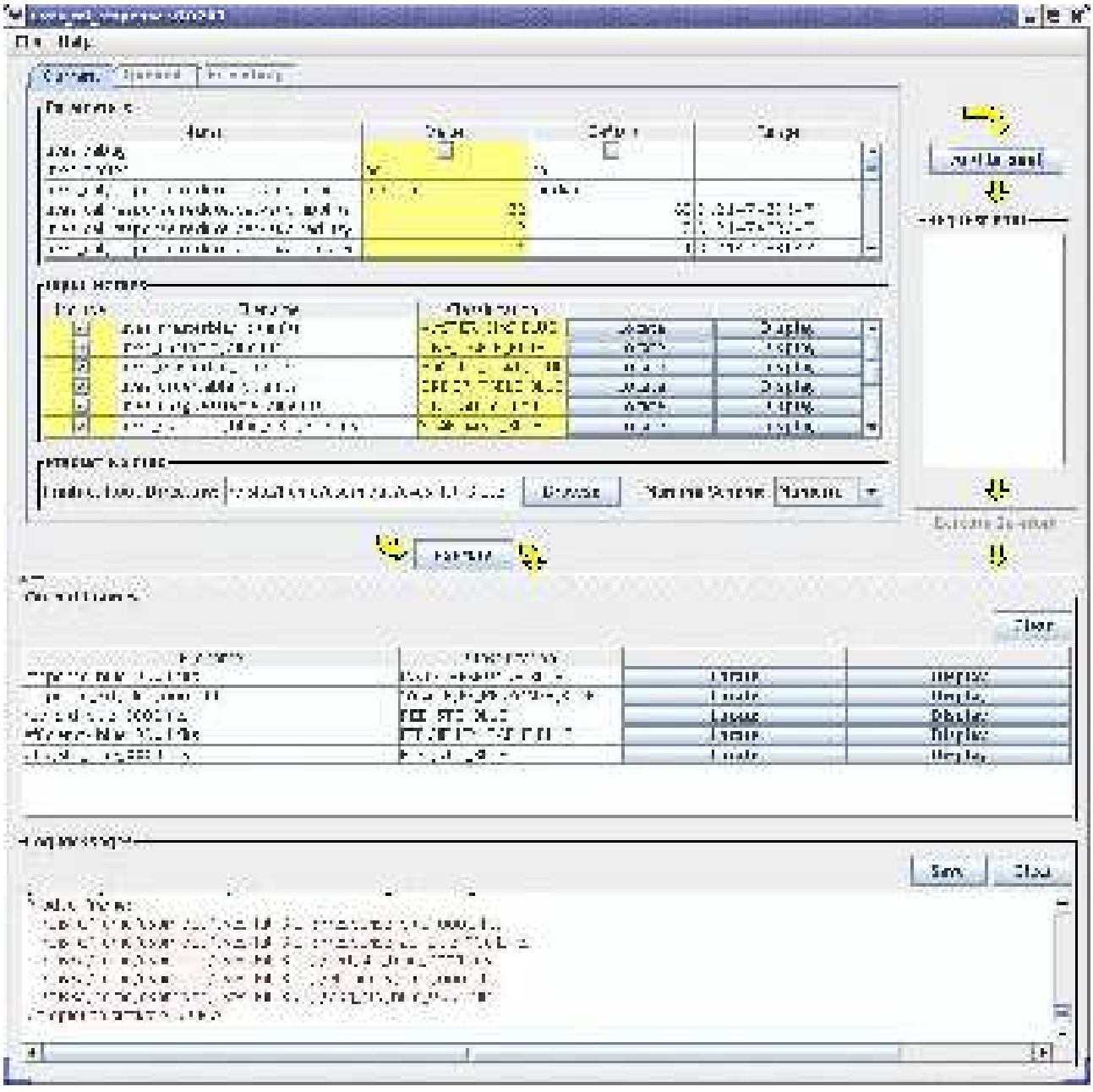


Figure 5.2.3: *The Gasgano recipe execution window.*

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 34). 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 [2] in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [3] DO categories for the supported UVES input raw frames are indicated in section 7.9.

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

```
/path_raw/UVES.2004-08-14T10:20:56.497.fits  ARC_LAMP_BLUE
/path_pro/ordertable_blue.fits               ORDER_TABLE_BLUE
/path_pro/lineguess_blue.fits                LINE_GUESS_TAB_BLUE
/path_pro/orderguess_blue.fits               ORDER_GUESS_TAB_BLUE
/path_pro/master_bias_blue.fits              MASTER_BIAS_BLUE (optional)
/path_pro/master_flat_blue.fits              MASTER_FLAT_BLUE (optional)
/path_ref/thargood_3.fits                    LINE_REFER_TABLE
/path_ref/line_intmon.fits                   LINE_INTMON_TABLE (for operations)
```

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 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 *uves_cal_wavecal* will treat the frame `/path_raw/UVES.2004-08-14T10:20:56.497.fits` as an ARC_LAMP_BLUE, the frame `/path_pro/ordertable_blue.fits` as a ORDER_TABLE_BLUE, 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 corresponding SOF example for UVES RED arm data would look like the following:

```
/path_raw/UVES.2004-08-16T12:24:16.345.fits  ARC_LAMP_RED
/path_pro/ordertable_redl.fits               ORDER_TABLE_REDL
/path_pro/lineguess_redl.fits                LINE_GUESS_TAB_REDL
/path_pro/orderguess_redl.fits               ORDER_GUESS_TAB_REDL
/path_pro/master_bias_redl.fits              MASTER_BIAS_REDL (optional)
/path_pro/master_flat_redl.fits              MASTER_FLAT_REDL (optional)
/path_pro/ordertable_redu.fits               ORDER_TABLE_REDU
/path_pro/lineguess_redu.fits                LINE_GUESS_TAB_REDU
/path_pro/orderguess_redu.fits               ORDER_GUESS_TAB_REDU
/path_pro/master_bias_redu.fits              MASTER_BIAS_REDU (optional)
/path_pro/master_flat_redu.fits              MASTER_FLAT_REDU (optional)
/path_ref/thargood_3.fits                    LINE_REFER_TABLE
/path_ref/line_intmon.fits                   LINE_INTMON_TABLE (for operations)
```

[2] 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*

[3] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 5.2.3, page 22).

The reason of this lack of control is that the 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 18). 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:

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.[4] 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 *uves_cal_wavecal* has its *EsoRex* generated configuration file named `uves_cal_wavecal.rc`, and is generated with the command:

esorex --create-config uves_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.
uves_cal_wavecal.calibrate.tolerance=0.6
```

[4] The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 5.2.3, page 22).

In this example, the parameter `uves_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**[5].

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

esorex --recipe-config=my_alternative_recipe_config

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 *uves_cal_wavecal* for processing the files specified in the set-of-frames `uves_cal_wavecal.sof`:

esorex uves_cal_wavecal uves_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 *uves_cal_wavecal* recipe *tolerance* parameter to `0.07`, the following should be typed:

esorex uves_cal_wavecal --tolerance=0.07 uves_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 uves_cal_wavecal --debug uves_cal_wavecal.sof

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

esorex uves_cal_wavecal --plotter='gnuplot -persist' uves_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 UVES BLUE arm. The data reduction of data obtained with the red arm is similar. The user will provide raw data with similar tag but the suffix BLUE will be replaced by RED, and twice as much calibration data for the lower and the upper chip where having similar tags but suffix REDL and REDU instead of BLUE.

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

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_blue1.fits    BIAS_BLUE
/path_raw/uves_bias_blue2.fits    BIAS_BLUE
/path_raw/uves_bias_blue3.fits    BIAS_BLUE
/path_raw/uves_bias_blue4.fits    BIAS_BLUE
/path_raw/uves_bias_blue5.fits    BIAS_BLUE
/path_raw/uves_dark_blue1.fits    DARK_BLUE
/path_raw/uves_dark_blue2.fits    DARK_BLUE
/path_raw/uves_dark_blue3.fits    DARK_BLUE
/path_raw/uves_flat_blue1.fits    FLAT_BLUE
/path_raw/uves_flat_blue2.fits    FLAT_BLUE
/path_raw/uves_flat_blue3.fits    FLAT_BLUE
/path_raw/uves_flat_blue4.fits    FLAT_BLUE
/path_raw/uves_flat_blue5.fits    FLAT_BLUE
/path_raw/uves_arc_form_blue.fits ARC_FORM_BLUE
/path_raw/uves_oflat_blue.fits    ORDER_FLAT_BLUE
/path_raw/uves_arc_lamp_blue.fits ARC_LAMP_BLUE
/path_raw/uves_standard_blue.fits STANDARD_BLUE
/path_raw/uves_science_blue.fits  SCIENCE_BLUE
/path_ref/thargood_3.fits         LINE_REFER_TABLE
/path_ref/flxstd.fits             FLUX_STD_TABLE
/path_ref/atmoexan.fits           EXTCOEFF_TABLE
```

Then run

esorex uves_obs_redchain uves_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 uves_cal_mflat will be executed. If the user already has a good master frame (master bias, master dark or master 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, uves_cal_mflat) will not be executed.

In the following a typical step-by-step data reduction procedure is described. [6] Figure 5.3.1 gives an overview.

We suggest the user to group the data according to detector arm, dichroic and detector binning setting. In the examples below we suppose the user has data acquired with the BLUE arm. `/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',

```
/path_raw/uves_arc_lamp_form_blue.fits ARC_LAMP_FORM_BLUE
```

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

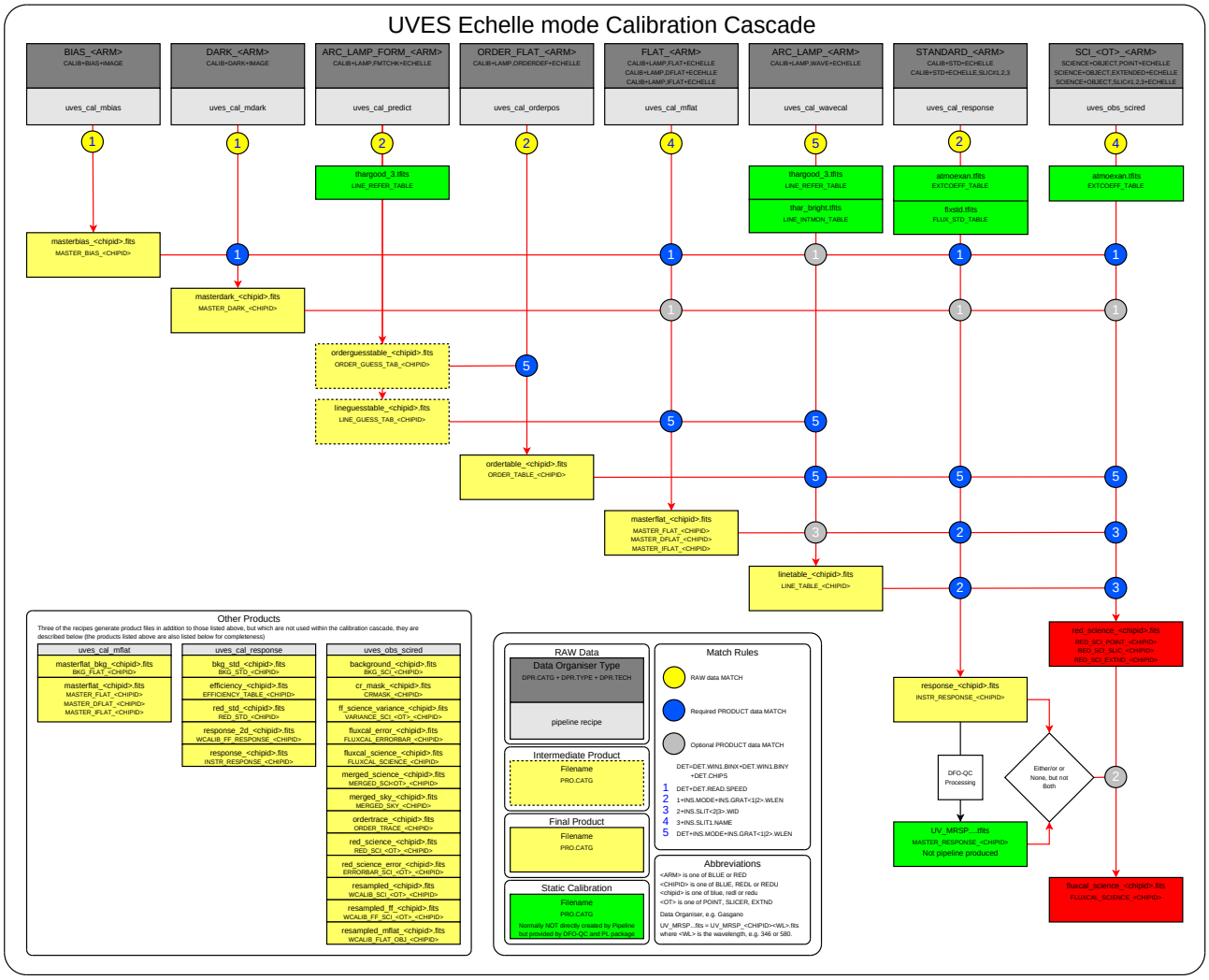


Figure 5.3.1: *The UVES calibration cascade.*

Order tracing flat field frames: these frames are characterized by DPR.TYPE=’LAMP,ORDERDEF’

```
/path_raw/uves_order_flat_blue.fits ORDER_FLAT_BLUE
orderguesstable_blue.fits           ORDER_GUESS_TAB_BLUE
```

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

```
/path_raw/uves_arc_lamp_blue.fits ARC_LAMP_BLUE
```

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

```
/path_raw/uves_bias_blue1.fits  BIAS_BLUE
/path_raw/uves_bias_blue2.fits  BIAS_BLUE
/path_raw/uves_bias_blue3.fits  BIAS_BLUE
/path_raw/uves_bias_blue4.fits  BIAS_BLUE
/path_raw/uves_bias_blue5.fits  BIAS_BLUE
```

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

```
/path_raw/uves_flat_blue1.fits FLAT_BLUE
/path_raw/uves_flat_blue2.fits FLAT_BLUE
/path_raw/uves_flat_blue3.fits FLAT_BLUE
/path_raw/uves_flat_blue4.fits FLAT_BLUE
/path_raw/uves_flat_blue5.fits FLAT_BLUE
```

To have additional information on the overall telescope and instrument response the user may want to reduce also telluric standard star frames: these frames are characterized by DPR.TYPE='STD'

```
/path_raw/uves_standard_blue.fits  STANDARD_BLUE
```

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

```
/path_raw/uves_science_blue.fits  SCIENCE_BLUE
```

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 [7].

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

   ```
   /path_raw/uves_arc_lamp_form_blue.fits      ARC_LAMP_FORM_BLUE
   /path_ref/thargood_3.fits                   LINE_REFER_TABLE
   ```

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

   esorex uves_cal_predict uves_cal_predict.sof

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

[7] 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_blue	table	LINE_GUESS_TAB_BLUE	guess line table
orderguesstable_blue	table	ORDER_GUESS_TAB_BLUE	guess order table

and generates the ASCII file lineguesstable_blue-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 uves_cal_orderpos.sof.

```
/path_raw/uves_order_flat_blue.fits ORDER_FLAT_BLUE
```

The user can generate an order table (PRO.CATG=ORDER_TABLE_blue) with the command:

esorex uves_cal_orderpos uves_cal_orderpos.sof

This command will generate the following products:

default recipe filename	format	PRO.CATG	short description
ordertable_blue	table	ORDER_TABLE_BLUE	order table

and generates the ASCII file ordertable_blue-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_blue1.fits  BIAS_BLUE
/path_raw/uves_bias_blue2.fits  BIAS_BLUE
/path_raw/uves_bias_blue3.fits  BIAS_BLUE
/path_raw/uves_bias_blue4.fits  BIAS_BLUE
/path_raw/uves_bias_blue5.fits  BIAS_BLUE
```

The command:

esorex uves_cal_mbias uves_cal_mbias.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
masterbias_blue	2d image (pix-pix)	MASTER_BIAS_BLUE	master bias

and generates the ASCII file masterbias_blue-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_blue1.fits  DARK_BLUE
/path_raw/uves_dark_blue2.fits  DARK_BLUE
/path_raw/uves_dark_blue3.fits  DARK_BLUE
/path_pro/masterbias_blue.fits  MASTER_BIAS_BLUE
```

The command:

esorex uves_cal_mdark uves_cal_mdark.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
masterdark_blue	2d image (pix-pix)	MASTER_DARK_BLUE	master dark

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

mv *.fits *.paf /path_pro

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

```
/path_raw/uves_flat_blue1.fits FLAT_BLUE
/path_raw/uves_flat_blue2.fits FLAT_BLUE
/path_raw/uves_flat_blue3.fits FLAT_BLUE
/path_pro/masterbias_blue.fits MASTER_BIAS_BLUE
/path_pro/ordertable_blue.fits ORDER_TABLE_BLUE
```

The command:

esorex uves_cal_mflat uves_cal_mflat.sof

will generate the following products:

default recipe filename	format	PRO.CATG	Note
masterflat_blue	2d image (pix-pix)	MASTER_FLAT_BLUE	master flat
masterflat_bkg_blue	2d image (pix-pix)	BKG_FLAT_BLUE	inter-order background

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

mv *.fits *.paf /path_pro

6. 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 uves_cal_wavecal.sof.

```
/path_raw/uves_arc_lamp_blue.fits   ARC_LAMP_BLUE
/path_pro/ordertable_blue.fits      ORDER_TABLE_BLUE
/path_pro/lineguesstable_blue.fits  LINE_GUESS_TAB_BLUE
/path_ref/thargood_3.fits           LINE_REFER_TABLE
/path_pro/masterbias_blue.fits      MASTER_BIAS_BLUE (optional)
/path_pro/masterflat_blue.fits      MASTER_FLAT_BLUE (optional)
```

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

The command

esorex uves_cal_wavecal uves_cal_wavecal.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
linetable_blue	table	LINE_TABLE_BLUE	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_blue'
Line table for trace 0, window #2 saved to extensions 4-6 of 'linetable_blue'
Line table for trace 0, window #3 saved to extensions 7-9 of 'linetable_blue'
```

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

mv *.fits *.paf /path_pro

7. As an optional step a standard star can be reduced in order to obtain the efficiency of the system telescope+instrument+detector. The std star raw frame will be stored in the following ASCII file uves_cal_response.sof together with previously obtained master calibration products:

```
/path_raw/uves_standard_blue.fits STANDARD_BLUE
/path_pro/ordertable_blue.fits    ORDER_TABLE_BLUE
/path_pro/linetable_blue.fits     LINE_TABLE_BLUE
/path_pro/masterbias_blue.fits    MASTER_BIAS_BLUE
/path_pro/masterdark_blue.fits    MASTER_DARK_BLUE (optional)
/path_pro/masterflat_blue.fits    MASTER_FLAT_BLUE
/path_ref/flxstd.fits             FLUX_STD_TABLE
/path_ref/atmoexan.fits           EXTCOEFF_TABLE
```

The command:

esorex uves_cal_response uves_cal_response.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
response_blue	1d (wav) image	INSTR_RESPONSE_BLUE	instrument response
response_2d_blue	2d (wav-order) image	WCALIB_FF_RESPONSE_BLUE	
red_std_blue	1d (wav) image	RED_STD_BLUE	reduced std star
efficiency_blue	table	EFFICIENCY_TABLE_BLUE	efficiency table
bkg_std_blue	2d (pix-pix) image	BKG_STD_BLUE	background frame of std star

and generates the ASCII file efficiency_blue-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 uves_obs_scired.sof:

```
/path_raw/uves_science_blue.fits SCIENCE_BLUE
/path_pro/ordertable_blue.fits   ORDER_TABLE_BLUE
/path_pro/linetable_blue.fits    LINE_TABLE_BLUE
/path_pro/masterbias_blue.fits   MASTER_BIAS_BLUE
/path_pro/masterdark_blue.fits   MASTER_DARK_BLUE (optional)
/path_pro/masterflat_blue.fits   MASTER_FLAT_BLUE
/path_pro/instrument_response_blue.fits   INSTR_RESPONSE_BLUE (optional)
/path_ref/atmoexan.fits           EXTCOEFF_TABLE (optional)
```

The frames tagged as INSTR_RESPONSE_BLUE and EXTCOEFF_TABLE are optional and used to eventually flux calibrate the reduced science spectra. If the user provides them, must provide both of them.

The command:

esorex uves_obs_scired uves_obs_scired.sof

will generate the following products:

default recipe file name	format	PRO.CATG	short description
red_science_blue	1d (wav) image	RED_SCIENCE_BLUE	extracted, flatfielded wavelength calibrated merged, sky subtracted science frame
merged_science_blue	1d (wav) image	MERGED_SCIENCE_BLUE	extracted, flatfielded wavelength calibrated merged science frame
resampled_science_blue	2d (wav-order) image	WCALIB_SCIENCE_BLUE	extracted, wavelength calibrated science frame
resampled_ff_science_blue	2d (wav-pix) image	WCALIB_FF_SCIENCE_BLUE	resampled flat field frame
resampled_mflat_blue	2d (wav-pix) image	WCALIB_FLAT_OBJ_BLUE	resampled flat frame
error_red_science_blue	1d (wav) image	ERRORBAR_SCIENCE_BLUE	error of reduced frame
variance_ff_science_blue	1d (wav) image	VARIANCE_SCIENCE_BLUE	variance of science frame
background_blue	2d (pix-pix) image	BKG_SCI_BLUE	background of science frame
ordertrace_blue	table	ORDER_TRACE_BLUE	order trace frame
cr_mask_blue	2d (pix-pix) image	CRMASK_BLUE	cosmic mask frame
merged_sky_blue	1d (wav) image	MERGED_SKY_BLUE	merged sky frame
fluxcal_science_blue	1d (wav) image	FLUXCAL_SCIENCE_BLUE	flux calibrated merged object frame
fluxcal_error_science_blue	1d (wav) image	FLUXCAL_ERRORBAR_SCIENCE_BLUE	error of fluxcalibrated merged science 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:

- Due to the difference in light paths between the flat-field lamp and the observed target the blaze function and interference fringe patterns are not fully corrected. At sufficiently high S/N this leaves residual patterns in the final spectrum at the scale of one order and at the scale of the inteference patterns.
- When flat-fielding in pixel-pixel space (non-default option), the statistical error bars from the flat-field are not propagated properly in the optimal extraction. If the object S/N is very high (comparable to the flat-field S/N) this causes a systematic bias of the final spectrum error bars. For this reason the default value **reduce.ffmethod=extract** is recommended.
- Line tilt correction (correcting for the fact that the dispersion relation depends on slit position) is not supported for 2d spectrum extractions, only for average/linear/optimal extractions.
- The recipe uves_obs_scired reduces only the first provided science frame. To reduce several science frames, the recipe should be run several times.
- If in the reduced spectra generated by the recipe uves_obs_scired are still present residual cosmic rays, we suggest the user to decrease the value of **uves.reduce.kappa**.
- Sometimes, using **uves_cal_wavecal.degree**=5, the user may found artifacts (absorption lines appearing as doublets) in the merged spectrum. For this reason we set parameter default value to 4. The user may also leave the pipeline find the best polynomial degree to have the best wavelength calibration accuracy. This is possible by setting a negative value for the **uves_cal_wavecal.degree** parameter.

7 Instrument Data Description

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

- Formatcheck
- Bias(es)
- Dark(s)
- Order definition
- Flat field(s)
- Wavelength calibration
- Reference standard star observation

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 with a ThAr lamp and a narrow slit. 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, for BLUE and RED arms separately, with the shutter closed. They are used to measure the dark current. They are measured for 1x1 and 2x2 binnings 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 Order definition frames

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

7.5 Flat Field frames

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 set of five frames 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.

7.6 Telluric lamp Flat Field frames

The so-called TFLATs are in fact internal, flat-field lamp flats. The principal difference between the TFLATs and normal FLATs is that the power supply of the TFLAT lamp is stabilised (hence TFLATs are sometimes also called 'stabilised flats').

The purpose of these flats is to monitor the performance of the instrument on a regular basis in a similar way as with spectrophotometric standard stars, but are complementary to the standard star observations with the advantages that internal lamp flats can be acquired during day time, isolate the instrument performance from that of the telescope and the transmission of the atmosphere and support various instrument configurations thus allowing to monitor the performance of the individual components of the instrument.

TFLATs are NOT suitable for scientific reductions.

7.7 Wavelength calibration frames

Wavelength calibration frames are long slit exposures taken (for UVES) with a ThAr arc lamp. They are used to find the wavelength calibration solution.

7.8 Reference standard star frames

Reference standard star calibrations are observations of standard stars for which the emitted spectra is known and which allows to determine the following:

- The response curve (i.e. the conversion between the science spectrum and a flux calibrated spectrum). The response curve provides a relative calibration flux which is better than the correction by the flat lamp (which corrects for the order-by-order blaze function, but still contains the lamp spectral slope).

- The overall efficiency (DQE) of telescope+instrument+detector (corrected for atmosphere extinction). This function can be evaluated for trending.

The observer will be able to reduce science data of the following types when using the described calibration frames:

- Point-like sources. The UVES pipeline has been designed with the purpose of being able to reduce this kind of source.
- Extended sources, by setting to the proper value the extraction method.
- Multi object sources. It is possible to reduce with some limitation more than one source on the slit whose spectra do not blend one with the other. The user can set the extraction slit size and position.

7.9 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 UVES templates (for a description of the 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`

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

Bias frames

- Template signature:
  `UVES_arm_cal_bias`
  `(x: blue, red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = BIAS`
- DO category:
  `BIAS_BLUE (NCHIP = 1)`
  `BIAS_RED (NCHIP = 2)`
- Pipeline recipe: `uves_cal_mbias`
- Relevant instrument parameters: group 1.
- Reference: Figure 7.9.0 (a).

Dark frames

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

Parasitic Light frames

- Template signature:
  `UVES_x_cal_flatfree`
  `(x: blue, red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = PARASITIC`
  `ESO DPR TECH = IMAGE`

- DO category:
  `PDARK_BLUE` (`NCHIP = 1`)
  `PDARK_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_cal_mdark`
- Relevant instrument parameters: group 1.

Order definition flat fields

- Template signature:
  `UVES_x_tec_orderdef`
  `(x: blue, red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,ORDERDEF`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `ORDER_FLAT_BLUE` (`NCHIP = 1`)
  `ORDER_FLAT_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_cal_orderpos`
- Relevant instrument parameters groups 1, 2.
- Reference: Figure 7.9.0 (c).

Spectroscopic flat fields

- Template signature:
  `UVES_x_cal_y`
  `(x: blue, red, dic1, dic2)`
  `(y: flatatt, flatfree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FLAT`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `FLAT_BLUE` (`NCHIP = 1`)
  `FLAT_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_cal_mflat`
- Relevant instrument parameters groups 1, 2.
- Reference: Figure 7.9.0 (e).

Deuterium lamp flat fields

- Template signature:
  `UVES_x_cal_y`
  `(x: blue, red, dic1, dic2)`
  `(y: flatfree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,DFLAT`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `DFLAT_BLUE` (`NCHIP = 1`)
  `DFLAT_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_cal_mflat`
- Relevant instrument parameters: groups 1, 2.

Absorption cell flat fields

- Template signature:
  `UVES_x_cal_y`
  `(x: blue, red, dic1, dic2)`
  `(y: flatfree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,IFLAT`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `IFLAT_BLUE` (`NCHIP = 1`)
  `IFLAT_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_cal_mflat`
- Relevant instrument parameters: groups 1, 2.

Format check spectra

- Template signature:
  `UVES_x_tec_fmtchk`
  `(x: blue, red, dic1, dic2)`

- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FMTCHK`
  `ESO DPR TECH = ECHELLE`

- DO category:
  `ARC_LAMP_FORM_BLUE(NCHIP = 1)`
  `ARC_LAMP_FORM_RED (NCHIP = 2)`

- Pipeline recipe: `uves_cal_predict`

- Relevant instrument parameters: groups 1, 2.

- Reference: Figure 7.9.0 (b).

Cross disperser alignment check spectra

- Template signature:
  `UVES_x_cal_cd_align`
  `(x:  blue, red, dic1, dic2)`

- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,CDALIGN`
  `ESO DPR TECH = SPECTRUM`

- DO category:
  `CD_ALIGN_BLUE(NCHIP = 1)`
  `CD_ALIGN_RED (NCHIP = 2)`

- Pipeline recipe: `uves_cal_cdalign`

- Relevant instrument parameters: groups 1, 2.

Wavelength calibration spectra

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

- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,WAVE`
  `ESO DPR TECH = ECHELLE`

- DO category:
  `ARC_LAMP_BLUE` (`NCHIP = 1`)
  `ARC_LAMP_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_cal_wavecal`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.9.0 (d).

Standard star spectra

- Template signatures:
  `UVES_x_obs_y`
  `(x: blue, red, dic1, dic2)`
  `(y: std, stdfree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = STD`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `STANDARD_BLUE` (`NCHIP = 1`)
  `STANDARD_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_cal_response`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.9.0 (f).

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 = ECHELLE`
- DO category:
  `SCIENCE_BLUE` (`NCHIP = 1`)
  `SCIENCE_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_obs_scired`
- Relevant instrument parameters: groups 1, 2.

Science spectra with image slicers

- Template signatures:
  `UVES_x_acq_imsl`
  `(x: blue, red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = SCIENCE`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `SCIENCE_BLUE` (`NCHIP = 1`)
  `SCIENCE_RED` (`NCHIP = 2`)
- Pipeline recipe: `uves_obs_scired`
- Relevant instrument parameters: groups 1, 2,

Slicer Number:	`ESO INS SLIT1 NAME`
Number of Slices:	`ESO INS SLIT1 NO`

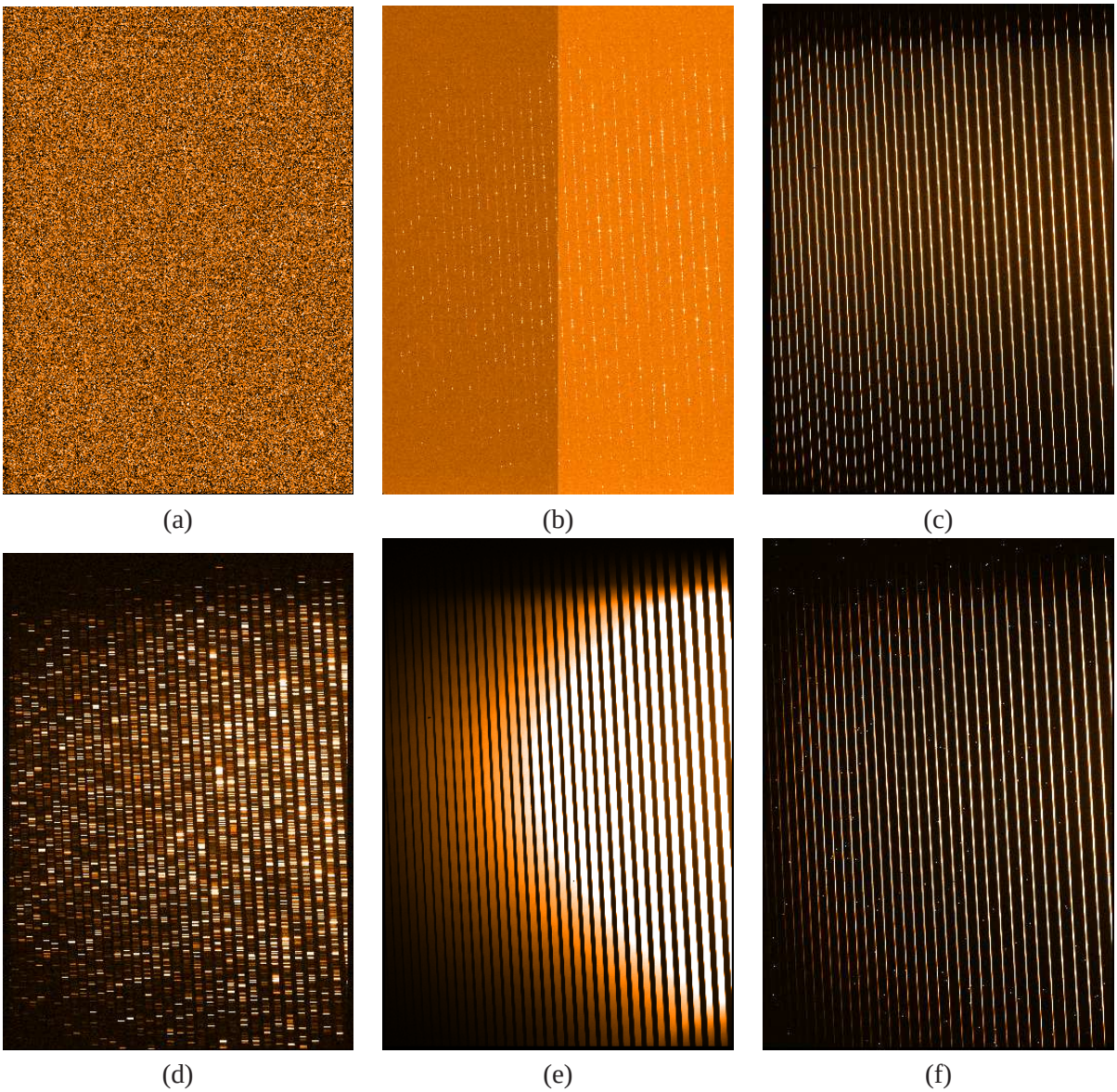


Figure 7.9.0: (a) a raw bias frame; (b) a raw formatcheck frame; (c) a raw order frame; (d) a raw arc lamp frame; (e) a raw flatfield frame; (f) a raw std star frame.

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 23). 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.

8.2 Standard stars flux table

This table contains the expected flux calibrated spectra for a given list of standard stars. Its PRO.CATG is FLUX_STD_TABLE.

8.3 Atmospheric extinction table

This table provides the extinction coefficient as a function of the wavelength expressed in Angstroem. It was originally compiled by Michel Rosa and embedded into MIDAS using a three component model that includes the effect of Rayleigh scattering, aerosols and ozone.

The description of that model (as fitted to data of La Silla) can be found in (Tueg, H., 1980, Astron.Astrophys. 82, 195), which makes reference to the generic model of Hayes & Latham (1975, ApJ 197, 593).

Its PRO.CATG is EXTCOEFF_TABLE.

8.4 Table to monitor line intensity

For quality control purposes a table listing reference lines of intermediate intensity uniformely distributed on the blue/red arms can be provided by DFO and has PRO.CATG equal to LINE_INTMON_TABLE.

8.5 Table to flux calibrate science data

In order to flux calibrate the science data an instrument response curve must be provided as input to the uves_obs_scired recipe. The response curve is either generated by the uves_cal_response recipe, or a master response curve (which is obtained by averaging individual response curves over a period of time) can be provided by DFO . Its PRO.CATG is INSTR_RESPONSE.

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 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 order need to be traced in a robust manner.
- Be able to compute and correct for detector pixel to pixel gain variations and the blaze function.
- Perform the wavelength calibration in a robust and automatic manner.
- Compute the telescope+instrument+detector efficiency.
- Reduce science data.

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 UVES pipeline involves the following input data:

- Raw frames:
  - Formatcheck frames to determine guess order and line tables.
  - Order definition frames to determine the order tables.
  - Arc lamp frames to determine the line table.
  - Bias frames to determine a master bias.
  - Dark frames to determine a master dark.
  - Flat frames to determine a master flat.
  - Standard star frames to compute the instrument response and the telescope+instrument+detector efficiency.
- calibration data products:
  - Guess order table to have a stable order tracing.
  - Guess line table to have an automatic, accurate and stable wavelength calibration.

  - Master bias, master dark to subtract them from master flat, standard star and science frames.
  - Master 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.
  - Instrument response to calibrate merged data in flux.

- Reference files:
  - Line table to produce guess and final line tables.
  - Atmospheric extinction table to be used for instrument efficiency computation.
  - Reference standard star fluxes to be used in instrument efficiency computation.

9.3 Reduction cascade

The UVES data reduction follows the sequence shows in figure 5.3.1.

In parenthesis we provide the DO category corresponding to each frame. The suffix CHIP can assume the following values: BLUE, REDL or REDU. The suffix ARM can assume the following values: BLUE or RED.

- Generate a guess order (ORDER_GUESS_TAB_CHIP) and a line tables (LINE_GUESS_TAB_CHIP) with the recipe **uves_cal_predict** from a formatcheck frame (ARC_LAMP_FORM_ARM) and a reference list of arc lines (LINE_REFER_TABLE).
- Refine the order guess table into an order table (ORDER_TABLE_CHIP) with the recipe **uves_cal_orderpos** from an order definition frame (ORDER_FLAT_ARM) obtained illuminating the UVES narrow slit with a continuum lamp.
- Build a master bias frame (MASTER_BIAS_CHIP) with the recipe **uves_cal_mbias** from a set of raw bias frames (BIAS_ARM) .
- Build a master dark frame (MASTER_DARK_CHIP) with the recipe **uves_cal_mdark** from a set of raw dark frames (DARK_ARM) and a master bias frame.
- Build a master flat frame (MASTER_FLAT_CHIP) with the recipe **uves_cal_mflat** from a set of raw flat frames (FLAT_ARM), a master bias and a master dark frame (if available). Additional input is the order table.
- Refine the guess line table into a line table (LINE_TABLE_CHIP) containing solution for three traces, the object trace and two sky traces, with the recipe **uves_cal_wavecal** from an arc lamp frame (ARC_LAMP_ARM) obtained illuminating the UVES wide slit with an arc lamp. Additional input are the order table and the reference line catalog (LINE_REFER_TABLE). We recommend also to use the master bias and the master flat.
- Determine the instrument response (INSTR_RESPONSE_CHIP) and telescope+instrument+detector efficiency (EFFICIENCY_CHIP) with the recipe **uves_cal_response** from a standard star observation (STANDARD_ARM) and a master bias, a master dark, a master flat, an order table, a line table, a table specifying the atmospheric extinction (EXTCOEFF_TABLE), and a table providing the standard star calibrated reference flux (FLUX_STD_TABLE).

- Reduce science data (SCIENCE_ARM) with the recipe **uves_obs_scired** using a master bias, a master dark, a master flat, an order table, a line table. To obtain flux calibrated spectra the user should also provide a table specifying the atmospheric extinction, and the instrument response curve.

Alternatively, to run the full data reduction chain, a user not willing to check results step-by-step may use the **uves_obs_redchain** recipe on a complete set of raw and ancillary data.

10 Pipeline Recipes Interfaces

In this section we provide for each recipe examples the required input data (and their classification tags). In the column labeled “nr” we indicate also the number of needed input frames with the following convention:

- 1 for a single frame (in case of pipeline calibration products, one for each detector chip of the corresponding arm),
- + for one or more input frames,
- ? for input frames which are optional (in case of pipeline calibration products one for each detector chip of the corresponding arm).

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_blue.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.

In the following we list examples using general file names and frame TAG values. For file names the indicated “arm” value have to be replaced either by “blue” or “red” while the frame tag name suffix “ARM” have to be replaced either by “BLUE” or “RED”. In case of pipeline product calibration frames, the “chip” value contained in the listed filenames, have to be replaced either by “blue” or, in case of data from the RED arm, by “redl” and “redu”. The corresponding frame tag suffix “CHIP” have to be replaced either by “BLUE” or by “REDL” and “REDU”, respectively for each pipeline calibration frame of the corresponding detector chip.

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 CATH	raw data category
HIERARCH ESO DPR TECH	raw data technique
HIERARCH ESO DET CHIPS	Instrument arm
HIERARCH ESO DET WIN1 BINX	Detector X binning
HIERARCH ESO DET WIN1 BINY	Detector Y binning
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 uves_cal_predict

The recipe uves_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

frame tag/category	nr	filename example
ARC_LAMP_FORM_ARM	1	/path_raw/uves_arc_lamp_form_arm.fits
LINE_REFER_TABLE	1	/path_ref/thargood_3.fits
MASTER_BIAS_CHIP	?	/path_pro/masterbias_chip.fits

10.1.2 Output

default recipe file name	format	PRO.CATG	short description
orderguesstable_chip.fits	table	ORDER_GUESS_TAB_CHIP	Guess order table
lineguesstable_chip.fits	table	LINE_GUESS_TAB_CHIP	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.DIFFXAV	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.DIFFYAV	Mean difference of predicted and measured line y positions
QC.MODEL.DIFFYMED	Median difference of predicted and measured line y positions
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

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 uves_cal_orderpos

The recipe uves_cal_orderpos generates an order table from a set of raw frames taken with a short length slit illuminated by a continuum lamp.

10.2.1 Input

frame tag/category	nr	filename example
`ORDER_FLAT_ARM`	1	`/path_raw/uves_order_flat_arm.fits`
`ORDER_GUESS_TAB_CHIP`	1	`/path_pro/orderguesstable_chip.fits`
`MASTER_BIAS_CHIP`	?	`/path_pro/masterbias_chip.fits`

10.2.2 Output

default recipe file name	format	PRO.CATG	short description
ordertable_chip.fits	table	ORDER_TABLE_CHIP	order table

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. A plot of the residuals is shown in Figure 11.2.4

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

frame tag/category	nr	filename example
`BIAS_ARM`	+	`/path_raw/uves_bias_arm.fits`

10.3.2 Output

default recipe file name	format	PRO.CATG	short description
masterbias_chip.fits	2d (pix-pix) image	MASTER_BIAS_CHIP	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.

```
--clean_traps         : Detector traps clean. If TRUE detector traps are
                        corrected.The bad pixels are replaced by the average
                        of nearest good pixels in the same column, or simply
                        marked as bad. The positions of bad pixels are
                        hard-coded (as function of UVES chip) their
                        intensities need to be corrected. [TRUE]
```

10.4 uves_cal_mdark

The recipe uves_cal_mdark computes a master dark frame.

10.4.1 Input

frame tag/category	nr	filename example
DARK_ARM	+	/path_raw/uves_dark_arm.fits
MASTER_BIAS_CHIP	1	/path_raw/masterbias_chip.fits

10.4.2 Output

default recipe file name	format	PRO.CATG	short description
masterdark_chip.fits	2d (pix-pix) image	MASTER_DARK_CHIP	master dark frame

10.4.3 Quality control

DFO trend the dark median level (PRO.DATAMED) for a given PRO.DATANCOM value and checks the values (min, max, median, mean, rms) of the dark on certain rectangular definable regions distributed across the detector.

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

```
--qc_dark.reg.num_x   : Number of regions on X direction (where mean/med/rms
                        are computed). [4]
--qc_dark.reg.num_y   : Number of regions on Y direction(where mean/med/rms
                        are computed). [4]
--qc_dark.reg.box_sx  : Region X size. [100]
--qc_dark.reg.box_sy  : Region Y size. [100]
--qc_dark.reg.border_x: X distance from the left hand side detector's border
                        and the left hand side regin's bottom corner. [100]
--qc_dark.reg.border_y: Y distance from the left hand side detector's border
                        and the left hand side regin's bottom corner. [100]
--qc_dark.reg.when    : When QC analysis is performed. 0: on each raw frame or
                        1: on the master frame. [0]
```

10.5 uves_cal_mflat

The recipe uves_cal_mflat computes the master flat frame.

10.5.1 Input

frame tag/category	nr	filename example
FLAT_ARM	+	/path_raw/uves_flat_arm.fits
MASTER_BIAS_CHIP	1	/path_raw/masterbias_chip.fits
ORDER_TABLE_CHIP	1	/path_pro/ordertable_chip.fits

10.5.2 Output

default recipe file name	format	PRO.CATG	short description
masterflat_chip.fits	2d (pix-pix) image	MASTER_FLAT_CHIP	master flat frame
masterflat_bkg_chip.fits	2d (pix-pix) image	BKG_FLAT_CHIP	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 uves_cal_mkmaster

The recipe uves_cal_mkmaster computes a master bias or a master dark or a master flat frame, depending on type (bias/dark/flat) of the provided input frames. It has been provided to match the corresponding recipe present in the MIDAS based release. Please refer to the recipes uves_cal_mbias, uves_cal_mdark, uves_cal_mflat for further description.

10.6.1 Input

Please refer to the corresponding sections of the recipes uves_cal_mbias, uves_cal_mdark, and uves_cal_mflat.

10.6.2 Output

Please refer to the corresponding sections of the recipes uves_cal_mbias, uves_cal_mdark, and uves_cal_mflat.

10.6.3 Quality control

Please refer to the corresponding sections of the recipes uves_cal_mbias, uves_cal_mdark, and uves_cal_mflat.

10.6.4 Parameters

Please refer to the corresponding sections of the recipes uves_cal_mbias, uves_cal_mdark, and uves_cal_mflat.

10.7 uves_cal_wavecal

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

10.7.1 Input

frame tag/category	nr	filename example
ARC_LAMP_ARM	1	/path_raw/uves_arc_lamp_arm.fits
ORDER_TABLE_CHIP	1	/path_pro/ordertable_chip.fits
LINE_GUESS_TAB_CHIP	1	/path_pro/lineguesstable_chip.fits
MASTER_BIAS_CHIP	1	/path_raw/masterbias_chip.fits
MASTER_FLAT_CHIP	1	/path_pro/masterflat_chip.fits
WEIGHTS_CHIP	1	/path_pro/weights_chip.fits
LINE_REFER_TABLE	1	/path_ref/thargood_3.fits
LINE_INTMON_TABLE	?	/path_ref/line_intmon.fits

10.7.2 Output

default recipe file name	format	PRO.CATG	short description
linetable_chip.fits	table	LINE_TABLE_CHIP	line table

The output line table(s), LINE_TABLE_CHIP, 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

The line FWHMs and the corresponding resolving power are measured along the dispersion direction (see also Figure 11.2.7). 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 11.2.6).

10.7.4 Parameters

parameter	alias	default
uves_cal_wavecal.nwindows	nwindows	3
uves_cal_wavecal.length	length	-1

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

10.7.5 Object-weighted extraction

Increased accuracy of the dispersion relation may be obtained by extracting the arclamp frame using the same weights as used for the science object optimal extraction. The necessary steps for this mode are (for blue arm data, red arm is similar):

- Run uves_cal_wavecal to produce a LINE_TABLE_CHIP. (This product is used only to bootstrap the next step, no particular accuracy is required.)
- Run uves_obs_scired with the --debug option set to true. This will create the FITS image file weights_chip.fits (which is the actual weight map used for the science object extraction) in the current directory.
- Re-run uves_cal_wavecal with the option --uves_cal_wavecal.extract.method set to 'weighted', and give as input the weights_chip.fits file, which must be classified as WEIGHTS_CHIP. This will generate the final LINE_TABLE_CHIP containing the dispersion solution based on the weighted arclamp extraction. [8]
- Re-run uves_obs_scired using the LINE_TABLE_CHIP generated in the previous step.

[8] For simplicity and consistency with the usual products of uves_cal_wavecal, this LINE_TABLE_CHIP contains the same dispersion solution three times.

10.8 uves_cal_response

10.8.1 Input

frame tag/category	nr	filename example
STANDARD_ARM	1	/path_raw/uves_standard_chip.fits
ORDER_TABLE_CHIP	1	/path_pro/ordertable_chip.fits
LINE_TABLE_CHIP	1	/path_pro/linetable_chip.fits
MASTER_BIAS_CHIP	1	/path_raw/masterbias_chip.fits
MASTER_DARK_CHIP	?	/path_raw/masterdark_chip.fits
MASTER_FLAT_CHIP	1	/path_raw/masterflat_chip.fits
FLUX_STD_TABLE	1	/path_ref/flxstd.fits
EXTCOEFF_TABLE	1	/path_ref/atmoexan.fits

10.8.2 Output

default recipe file name	format	PRO.CATG	short description
response_chip.fits	1d (wav) image	INSTR_RESPONSE_CHIP	instrument response
response_2d_chip.fits	2d (wav-order) image	WCALIB_FF_RESPONSE_CHIP	instrument response curve before correcting for exposure time, gain, binning and atmospheric absorption
red_std_chip.fits	1d (wav) image	RED_STD_CHIP	reduced std star
efficiency_chip.fits	table	EFFICIENCY_TABLE_CHIP	efficiency table
bkg_std_chip.fits	2d (pix-pix) image	BKG_STD_CHIP	background frame of std star

The efficiency table contains the following columns:

Wave	wavelength at blaze center
Eff	efficiency at blaze center
Binsize	size of bin
Order	relative order number

10.8.3 Quality control

The pipeline measures the following quality control parameters:

QC.ORDi.OBJ.SN	Average S/N at order center
QC.ORDi.OBJ.POS	Average OBJ POS at order center
QC.ORDi.OBJ.FWHM	Average FWHM on order
QC.EX.NORD	Number of orders extracted
QC.EX.XSIZE	Input image width (pixels)
QC.EX.YSIZE	Extraction slit (pixels)
QC.VRAD.BARYCOR	Barycentric radial velocity correction

QC.VRAD.HELICOR	Heliocentric radial velocity correction

where i is the order number. The parameters are logged directly in the reduced spectrum FITS file.

Additional QC parameters are logged in the efficiency table product:

QC.BLAZEFFi	Efficiency at blaze maximum
QC.BLAZWLENi	Wavelength at blaze maximum

See figure 11.2.9 for typical plots of the instrument response and the instrument efficiency curves.

10.8.4 Parameters

parameter	alias	default
uves_cal_response.reduce.backsub.mmethod	reduce.backsub.mmethod	median
uves_cal_response.reduce.backsub.npoints	reduce.backsub.npoints	82
uves_cal_response.reduce.backsub.radiusy	reduce.backsub.radiusy	2
uves_cal_response.reduce.backsub.sdegree	reduce.backsub.sdegree	1
uves_cal_response.reduce.backsub.smoothx	reduce.backsub.smoothx	-1.0
uves_cal_response.reduce.backsub.smoothy	reduce.backsub.smoothy	-1.0
uves_cal_response.reduce.extract.method	reduce.extract.method	optimal
uves_cal_response.reduce.extract.kappa	reduce.extract.kappa	10.0
uves_cal_response.reduce.extract.chunk	reduce.extract.chunk	32
uves_cal_response.reduce.extract.profile	reduce.extract.profile	auto
uves_cal_response.reduce.extract.skymethod	reduce.extract.skymethod	optimal
uves_cal_response.reduce.extract.oversample	reduce.extract.oversample	5
uves_cal_response.reduce.extract.best	reduce.extract.best	FALSE
uves_cal_response.reduce.slitlengt	reduce.slitlength	-1.0
uves_cal_response.reduce.skysub	reduce.skysub	TRUE
uves_cal_response.reduce.objoffset	reduce.objoffset	0.0
uves_cal_response.reduce.objslit	reduce.objslit	-1.0
uves_cal_response.reduce.ffmethod	reduce.ffmethod	extract
uves_cal_response.reduce.blazecorr	reduce.blazecorr	FALSE
uves_cal_response.reduce.rebin.wavestep	reduce.rebin.wavestep	-1.0
uves_cal_response.reduce.rebin.scale	reduce.rebin.scale	FALSE
uves_cal_response.reduce.merge	reduce.merge	optimal
uves_cal_response.efficiency.reduce.backsub.mmethod	efficiency.reduce.backsub.mmethod	median
uves_cal_response.efficiency.reduce.backsub.npoints	efficiency.reduce.backsub.npoints	82
uves_cal_response.efficiency.reduce.backsub.radiusy	efficiency.reduce.backsub.radiusy	2
uves_cal_response.efficiency.reduce.backsub.sdegree	efficiency.reduce.backsub.sdegree	1
uves_cal_response.efficiency.reduce.backsub.smoothx	efficiency.reduce.backsub.smoothx	-1.0
uves_cal_response.efficiency.reduce.backsub.smoothy	efficiency.reduce.backsub.smoothy	-1.0
uves_cal_response.efficiency.reduce.extract.method	efficiency.reduce.extract.method	linear
uves_cal_response.efficiency.reduce.extract.kappa	efficiency.reduce.extract.kappa	10.0

uves_cal_response.efficiency.reduce.extract.chunk	efficiency.reduce.extract.chunk	32
uves_cal_response.efficiency.reduce.extract.profile	efficiency.reduce.extract.profile	auto
uves_cal_response.efficiency.reduce.extract.skymethod	efficiency.reduce.extract.skymethod	optimal
uves_cal_response.efficiency.reduce.extract.oversample	efficiency.reduce.extract.oversample	5
uves_cal_response.efficiency.reduce.extract.best	efficiency.reduce.extract.best	TRUE
uves_cal_response.efficiency.reduce.slitlength	efficiency.reduce.slitlength	-1.0
uves_cal_response.efficiency.reduce.skysub	efficiency.reduce.skysub	TRUE
uves_cal_response.efficiency.reduce.objoffset	efficiency.reduce.objoffset	0.0
uves_cal_response.efficiency.reduce.objslit	efficiency.reduce.objslit	-1.0
uves_cal_response.efficiency.reduce.ffmethod	efficiency.reduce.ffmethod	no
uves_cal_response.efficiency.reduce.blazecorr	efficiency.reduce.blazecorr	FALSE
uves_cal_response.efficiency.reduce.rebin.wavestep	efficiency.reduce.rebin.wavestep	-1.0
uves_cal_response.efficiency.reduce.rebin.scale	efficiency.reduce.rebin.scale	FALSE
uves_cal_response.efficiency.reduce.merge	efficiency.reduce.merge	sum
uves_cal_response.efficiency.paccuracy	paccuracy	60.0

10.9 uves_obs_scired

This recipe reduces a science frame.

10.9.1 Input

frame tag/category	nr	filename example
SCIENCE_ARM	1	/path_raw/uves_science_chip.fits
ORDER_TABLE_CHIP	1	/path_pro/ordertable_chip.fits
LINE_TABLE_CHIP	1	/path_pro/linetable_chip.fits
MASTER_BIAS_CHIP	1	/path_raw/masterbias_chip.fits
MASTER_DARK_CHIP	?	/path_raw/masterdark_chip.fits
MASTER_FLAT_CHIP	1	/path_raw/masterflat_chip.fits
EXTCOEFF_TABLE	?	/path_ref/atmoexan.fits
INSTR_RESPONSE_CHIP	?	/path_pro/instrument_response_chip.fits

10.9.2 Output

default recipe file name	format	PRO.CATG	short description
red_science_chip	1d (wav) image	RED_SCIENCE_CHIP	extracted, flatfielded wavelength calibrated merged, sky subtracted science frame
merged_science_chip	1d (wav) image	MERGED_SCIENCE_CHIP	extracted, flatfielded wavelength calibrated merged science frame
resampled_science_chip	2d (wav-order) image	WCALIB_SCIENCE_CHIP	extracted, wavelength calibrated

			science frame
resampled_ff_science_chip	2d (wav-pix) image	WCALIB_FF_SCIENCE_CHIP	resampled flat field frame
resampled_mflat_chip	2d (wav-pix) image	WCALIB_FLAT_OBJ_CHIP	resampled flat frame
error_red_science_chip	1d (wav) image	ERRORBAR_SCIENCE_CHIP	error of reduced frame
variance_ff_science_chip	1d (wav) image	VARIANCE_SCIENCE_CHIP	variance of science frame
background_chip	2d (pix-pix) image	BKG_SCI_CHIP	background of science frame
ordertrace_chip	table	ORDER_TRACE_CHIP	order trace frame
cr_mask_chip	2d (pix-pix) image	CRMASK_CHIP	cosmic mask frame
merged_sky_chip	1d (wav) image	MERGED_SKY_CHIP	merged sky frame
fluxcal_science_chip	1d (wav) image	FLUXCAL_SCIENCE_CHIP	flux calibrated merged object frame
fluxcal_error_science_chip	1d (wav) image	FLUXCAL_ERRORBAR_SCIENCE_CHIP	error of fluxcalibrated merged science frame

In optimal extraction the reduced science frame is identical to the merged one. In average/linear extraction the reduced frame is sky subtracted, the merged not. WCALIB_* frames are rebinned frames not yet merged.

10.9.3 Quality control

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

QC.ORDi.OBJ.SN	Average S/N at order center
QC.ORDi.OBJ.POS	Average OBJ POS at order center
QC.ORDi.OBJ.FWHM	Average FWHM on order
QC.ORDi.OBJ.RPLPAR	Average relative ripple amplitude
QC.EX.NORD	Number of orders extracted
QC.EX.XSIZE	Input image width (pixels)
QC.EX.YSIZE	Extraction slit (pixels)
QC.VRAD.BARYCOR	Barycentric radial velocity correction
QC.VRAD.HELICOR	Heliocentric radial velocity correction

The parameters SN, POS, FWHM, RPLPAR give, order by order, an indication of the expected object signal to noise ratio, position at the center of the orders, FWHM, and extraction quality parameter (RPLPAR should be small, possibly less than 5). If the POS parameter reaches values close to half of the slit, probably due to a high value of the airmass, the object may fall outside the extraction slit. In those cases we suggest to use an order trace obtained by a standard star observation with the same airmass. If the object FWHM is comparable to the extraction slit, we recommend the user to use 2D extraction. If the ripple parameter is much higher than 5, we recommend the user to increase the parameter **reduce.extract.oversample** (this will proportionally increase the execution time), or eventually choose the extraction method linear or average.

10.9.4 Parameters

parameter	alias	default
uves_obs_scired.reduce.backsub.mmethod	reduce.backsub.mmethod	median
uves_obs_scired.reduce.backsub.npoints	reduce.backsub.npoints	82

uves_obs_scired.reduce.backsub.radiusy	reduce.backsub.radiusy	2
uves_obs_scired.reduce.backsub.sdegree	reduce.backsub.sdegree	1
uves_obs_scired.reduce.backsub.smoothx	reduce.backsub.smoothx	-1.0
uves_obs_scired.reduce.backsub.smoothy	reduce.backsub.smoothy	-1.0
uves_obs_scired.reduce.extract.method	reduce.extract.method	optimal
uves_obs_scired.reduce.extract.kappa	reduce.extract.kappa	10.0
uves_obs_scired.reduce.extract.chunk	reduce.extract.chunk	32
uves_obs_scired.reduce.extract.profile	reduce.extract.profile	auto
uves_obs_scired.reduce.extract.skymethod	reduce.extract.skymethod	optimal
uves_obs_scired.reduce.extract.oversample	reduce.extract.oversample	5
uves_obs_scired.reduce.extract.best	reduce.extract.best	TRUE
uves_obs_scired.reduce.slitlength	reduce.slitlength	-1.0
uves_obs_scired.reduce.skysub	reduce.skysub	TRUE
uves_obs_scired.reduce.objoffset	reduce.objoffset	0.0
uves_obs_scired.reduce.objslit	reduce.objslit	-1.0
uves_obs_scired.reduce.ffmethod	reduce.ffmethod	extract
uves_obs_scired.reduce.blazecor	reduce.blazecorr	FALSE
uves_obs_scired.reduce.rebin.wavestep	reduce.rebin.wavestep	-1.0
uves_obs_scired.reduce.rebin.scale	reduce.rebin.scale	FALSE
uves_obs_scired.reduce.merge	reduce.merge	optimal
uves_obs_scired.reduce.merge_delt1	reduce.merge_delt1	0.0
uves_obs_scired.reduce.merge_delt2	reduce.merge_delt2	0.0

10.10 uves_obs_redchain

This recipe performs the full UVES data reduction.

10.10.1 Input

frame tag/category	nr	filename example
BIAS_ARM	+	/path_raw/uves_bias_arm.fits
DARK_ARM	+	/path_raw/uves_dark_arm.fits
FLAT_ARM	+	/path_raw/uves_flat_arm.fits
ARC_LAMP_FORM_ARM	1	/path_raw/uves_arc_form_arm.fits
ORDER_FLAT_ARM	1	/path_raw/uves_oflat_arm.fits
ARC_LAMP_ARM	1	/path_raw/uves_arc_lamp_arm.fits
STANDARD_ARM	1	/path_raw/uves_standard_arm.fits
SCIENCE_ARM	1	/path_raw/uves_science_arm.fits
LINE_REFER_TABLE	1	/path_ref/thargood_3.fits
FLUX_STD_TABLE	1	/path_ref/flxstd.fits
EXTCOEFF_TABLE	1	/path_ref/atmoexan.fits

10.10.2 Output

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

10.10.3 Quality control

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

10.10.4 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

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

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 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.

11.1.3 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**.

11.1.4 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**).

11.1.5 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.

11.1.6 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.

11.1.7 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: $\mathbf{tolerance} < \mathbf{ALPHA} * |\lambda\mathbf{\_cat} - \lambda\mathbf{\_nn}|$. 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.

11.1.8 Optimal extraction

The pipeline uses optimal extraction to achieve higher signal-to-noise (S/N) of faint objects, corresponding to an increase in effective exposure time up to 70% compared with a simple aperture extraction (see Horne [17] and the further developments by Marsh [18] and Mukai [19] for an introduction).

In order to measure the spatial profile, a preliminary estimate and subtraction of the sky is carried out by taking the median of all pixels (after masking out the object using a rough object localization). The algorithm then follows Horne's scheme but with the following differences:

- The spatial profile is measured either using an analytical (Gaussian or Moffat) profile as described in [19], or by resampling the empirical profile to a grid with a resolution of 0.2 pixel in the spatial direction, and fitting a low order polynomial to the spatial profile at each resampled position. [9] The parameter **reduce.extract.profile** defines the kind of profile to use. See also figure 11.1.3.
- In order to fully exploit the peculiarities of the echelle format the free parameters of the respective models (analytical profile centroid and width, or virtually resampled profile at each spatial position) are modelled as 2D polynomials in wavelength *and* order number. In this way, regions (which may span entire orders) where the profile cannot be accurately determined due to very low signal are interpolated from neighbouring regions having presumably higher S/N.
- Horne's formula for the optimally extracted flux (which is equivalent to profile fitting at every wavelength [18]) assumes that the sky background has been already subtracted, and furthermore that the interpolated sky level is effectively noise-free. Because of the short slits typically used in echelle spectroscopy (to ensure order separation), the assumption about a noise-free sky determination may not be valid; we therefore generalized the method to give combined optimal sky and object flux estimates by minimization of

$$\chi^2 = \sum_i \frac{(f_i - (S_i + F p_i))^2}{\sigma_i^2} \tag{2}$$

where f_i and σ_i^2 are the flux and variance at the i'th pixel, p_i is the normalized spatial profile, and F and S_i are the object flux and sky levels to be determined. Assuming a simple model where the sky background is constant, $S_i = S$, a two-parameter minimization of eq. (2) yields

$$F = \frac{(\sum_i 1/\sigma_i^2)\sum_i p_i f_i/\sigma_i^2 - (\sum_i p_i/\sigma_i^2)\sum_i f_i/\sigma_i^2}{D} \tag{3}$$

$$S = \frac{(\sum_i p_i^2/\sigma_i^2)\sum_i f_i/\sigma_i^2 - (\sum_i p_i/\sigma_i^2)\sum_i p_i f_i/\sigma_i^2}{D} \tag{4}$$

with variances

$$V(F) = \frac{\sum_i 1/\sigma_i^2}{D}; \qquad V(S) = \frac{\sum_i p_i^2/\sigma_i^2}{D}, \tag{5}$$

[9] While resampling the data is often avoided because it introduces resampling noise [18], this resampling noise is smoothed when fitting low-degree polynomials to the spatial profile. However, when the model profile is later used to extract the data, it is important that the model is rebinned to the sampling of the data rather than the other way around. Mukai dubbed this "virtual resampling".

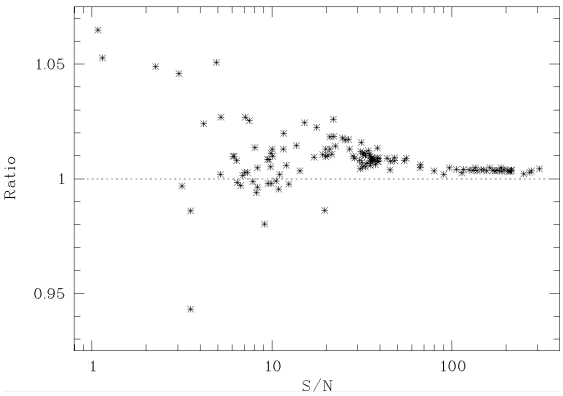


Figure 11.1.1: Ratio of the S/N obtained with the optimal sky subtraction and the S/N obtained with a separate, initial sky subtraction. Each point in the figure corresponds to the extraction of one echelle order in various data sets with different count levels.

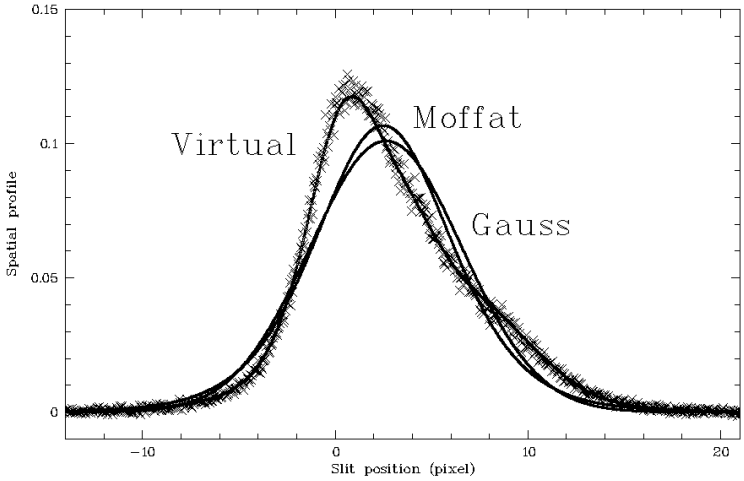


Figure 11.1.2: Empirical spatial profile (crosses) and three models (lines). At intermediate or high S/N the analytical methods usually cannot fit the object spatial profile accurately. (The UVES instrumental PSF is known to be slightly asymmetric, but the high degree of asymmetry in the present data set, chosen here for illustration purposes, is due to the source being double.)

where $D = (\sum_i 1/\sigma_i^2) \sum_i p_i^2/\sigma_i^2 - (\sum_i p_i/\sigma_i^2)^2$, and where the error bars of p_i and σ_i^2 are not propagated because the final variances are dominated by the contribution from f_i (following Horne). The object and sky spectra are finally extracted by applying eqs. (3) and (4) to the non-sky subtracted image. In comparison with a separate sky subtraction, this method improves the final object S/N by a few percent (see figure 11.1.1) in the low to intermediate S/N range. While this is not a dramatic improvement, we consider it worth the effort because the additional computational cost is practically zero.

Cosmic ray detection As in Horne's algorithm pixels polluted by cosmic ray hits and CCD defects are identified by the relation

$$|f_i - (S + Fp_i)| > \kappa\sigma_i$$

where κ is equal to the recipe parameter **reduce.extract.kappa**

Robust automatic data reduction Considering the default resolution of 0.2 pixel, a 30 pixels slit length and a two-dimensional 2nd degree polynomial, the virtual method needs to determine 1350 polynomial coefficients from the science image. In contrast, the analytical methods use only 18 polynomial coefficients for the two 2nd degree fits to the centroid and width of the spatial profile. For this reason, the analytical methods are more robust at very low S/N (see figure 11.1.4). However, at higher S/N the often significant mismatch between the analytical profile and the empirical profile (see figure 11.1.2) is known to bias the extracted flux [18] and make the rejection of cosmic rays unreliable.

In order to have a fully automatic data reduction with always good quality science results, the appropriate profile measuring method is selected at runtime, depending on an initial estimate of the object S/N. If it is less than 10,

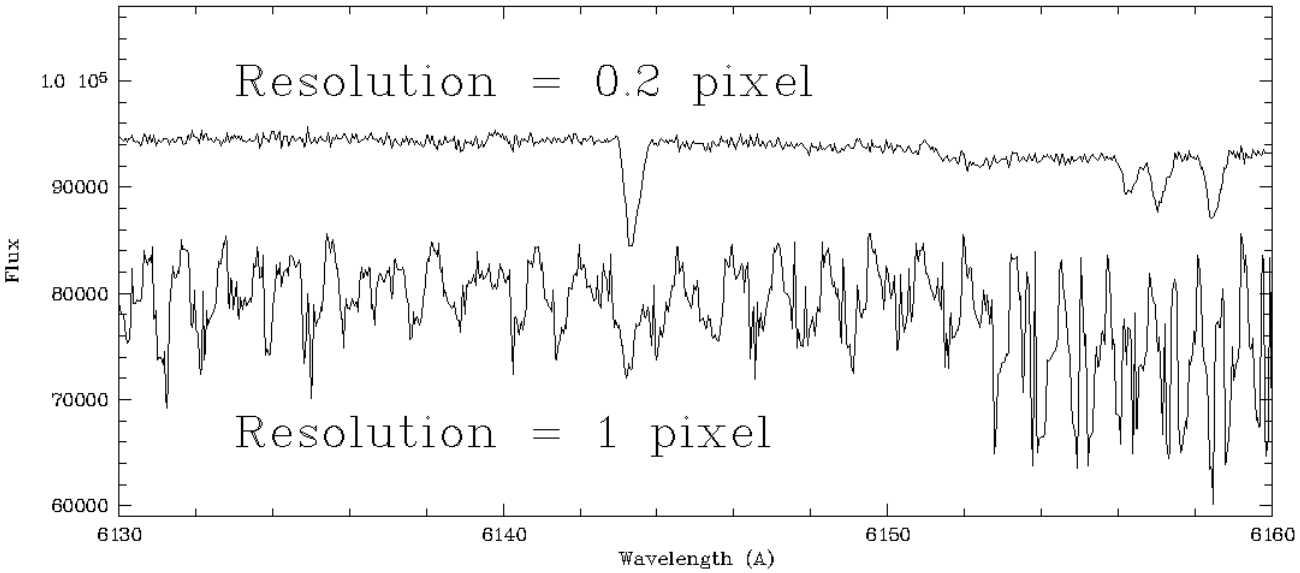


Figure 11.1.3: Zoom of spectrum with S/N $\approx$ 290. Using a model profile of too low resolution may cause a quasi-periodic pattern in the extracted spectrum.

an analytical method is used, otherwise the virtual method is used. Additionally, if the estimated S/N is > 200, the resolution is decreased from 0.2 pixel to 0.1 pixel. The Gauss and Moffat methods usually give very similar results, and the Gauss method is chosen as default because it makes the data reduction slightly faster.

The virtual method assumes that the object spatial profile varies slowly with wavelength. This might not be true if the object offset varies significantly with respect to the order location defined by the uves_cal_orderpos recipe. Therefore the order definition step is initially repeated, but using the science frame.

Also, the degrees of all two-dimensional, low-order polynomials involved are determined at runtime by starting from (0, 0) and increasing the degrees in steps of one or two as long as the residuals decrease significantly. All polynomial fits are weighted, and outliers are rejected iteratively using a robust kappa-sigma clipping. Although this method of determination of the optimal polynomial degrees involves some ad hoc heuristics, it has proven to consistently give better results than setting the degrees manually.

Assessing the extraction quality The default values of the optimal extraction recipe parameters were chosen (and adjusted at runtime as necessary) in order to achieve a good extraction quality over a wide range of input data.

However, to assess the quality of the optimal extraction, the relevant recipe should be run in debugging mode (by setting the parameter **debug** to TRUE). This will create two intermediate FITS files b_blue.fits and simulate_blue.fits in the directory where EsoRex or Gasgano was launched (see figures 11.1.5 and 11.1.6). b_blue.fits is the pre-processed (i.e. rotated, bias corrected, interorder background subtracted) input frame as provided to the extraction algorithm. simulate_blue.fits is a reconstructed image, simulated from the extracted object+sky spectra and cosmic ray map as well as the measured spatial profile. The reconstructed image thereby shows how the extraction algorithm was able to measure the spatial profile, exactly which cosmic rays were detected etc. and allows finetuning the recipe parameters so that a close resemblance to the input frame is achieved.

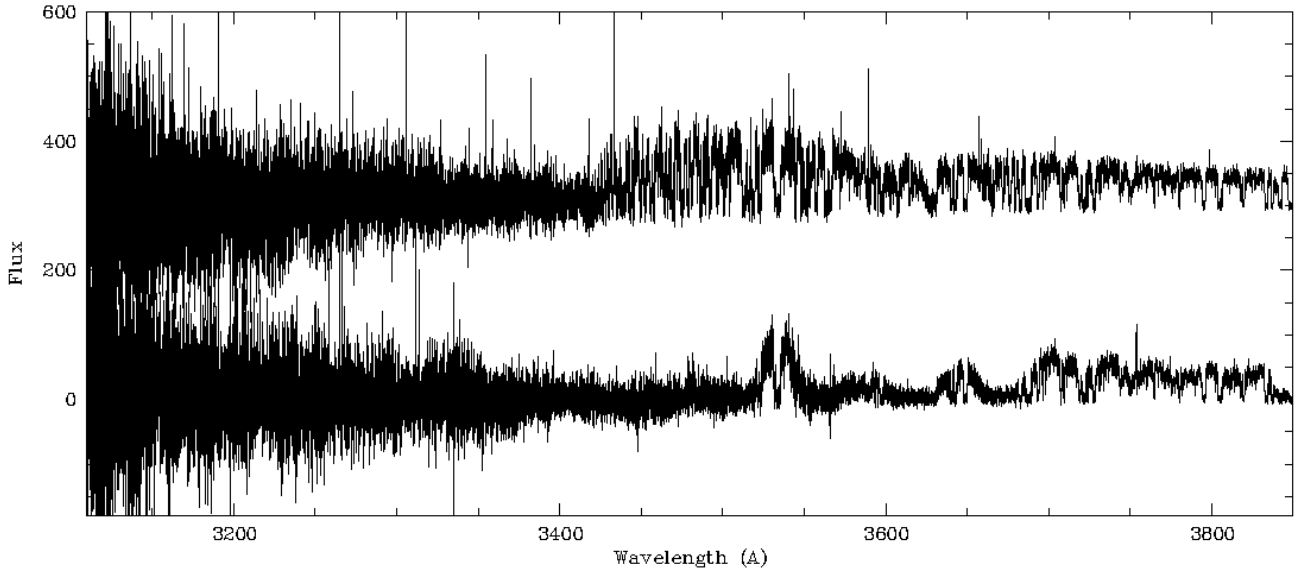


Figure 11.1.4: Comparison of an extracted QSO spectrum obtained with the new algorithm, analytical method (upper spectrum, shifted +300 units for clarity) with the same spectrum as produced using the previous algorithm (lower spectrum). The latter presents regions of missing signal because the object tracing fails in some orders. The global model used in the current implementation allows to locate and extract the object in every order in a robust manner.

The reconstructed image is produced also in average/linear extraction. In this case the image would have no information about cosmic rays and the spatial profile, which are not measured in average/linear extraction modes.

11.1.9 Linear extraction.

In linear extraction, for each order, for each wavelength the object pixel intensities within the given extraction slit length are summed. If the extraction slitlength and offset are not specified by the user (default), the object offset is measured (using a simple centroid) and appropriate object and sky extraction windows are defined.

11.1.10 Average extraction.

The average extraction is the same as the linear extraction, but the resulting flux is divided by the slit length in pixels.

11.1.11 Spatial extraction.

The so called spatial or 2D extraction is a spectrum extraction algorithm which keeps the full spatial spectrum resolution. It is indicated for the extraction of object spectra whose cross order FWHM is of the order of 1/3 or more of the instrument observation slit length, or in case that more than one object spectrum is observed along the slit length and the object spectra are blended one with another.

Figure 11.1.5: Input science image (detail).

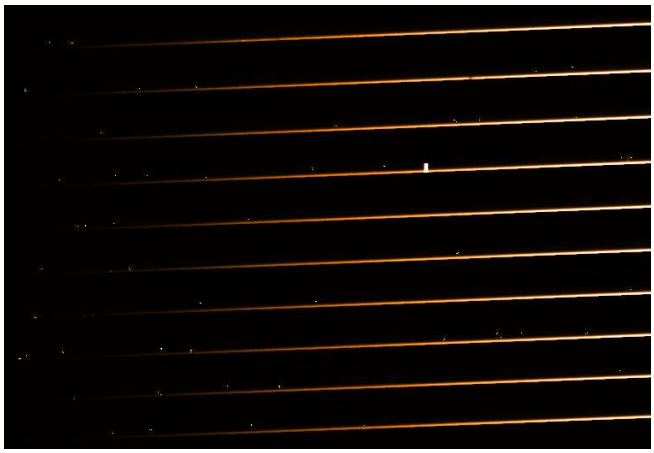


Figure 11.1.6: Science image reconstructed from the extracted spectra and cosmic ray map. For illustration purposes only the spectrum was extracted using a short slit (1/4 of full slit) with a small positive offset (+1/8 of full slit). This reconstructed image shows the locations of the detected cosmic rays, the sky emission line, and that the overall spectrum shape is very similar to the input spectrum. Of course, regions outside the specified extraction slit are not “seen” by the extraction algorithm and are therefore black.

This extraction is activated by setting the parameter **reduce.extract.method** to 2d. Initially if **reduce.ffmethod** is set to "pixel", the object spectrum is divided by the flat field.

In this case after spectrum re-orientation, the orders are rotated and realigned by applying a 2D transformation controlled by the order trace polynomial coefficients.

If **reduce.ffmethod** is set to "extract", the re-oriented object image is divided by the re-oriented flat field image. If **reduce.ffmethod** is set to "no", no division by the flat field is performed.

Then the spectrum is rebinned at a sampling step set by the parameter **reduce.rebin.wavestep** and eventually properly rescaled in flux according to the value of the parameter **reduce.rebin.scale**. Finally different orders are merged using one of the two possible methods "sum" (simple summation of flux in adjacent orders, appropriate for non-flatfielded spectra) or "optimal" (average weigthed by the inverse flux variances, appropriate for flatfielded spectra) defined by the parameter **reduce.merge**.

11.1.12 Reduction of more than an object on the slit

The UVES pipeline has been designed to reduce single point-like objects on the slit but has basic support for extracting more than one object on the slit. If the object spectra are not blended the user can use iteratively optimal or average extraction and set the slit offset and size to get the best extraction quality of each object. The extraction slit length and offset are defined by the parameters **reduce.slitlength** and **reduce.objoffset** respectively.

11.1.13 Efficiency computation

The recipe uves_cal_response computes the instrument response curve as well as the overall wavelength dependent efficiency of the telescope+instrument+detector by reducing a standard star spectrum.

The instrument response is calculated as the ratio between the standard star flux and the extracted+resampled spectrum, after correcting the reduced spectrum for atmospheric extinction (depending on the airmass). The tabulated flux is resampled to the standard star spectrum sampling by 3rd order spline Hermite interpolation.

After computation of the effective counts per ergs ($ctsperg$) the following formula is used:

$$eff = \frac{ccd\_gain}{nphot},$$

where

$$nphot = \frac{1}{ctsperg} \cdot erg2phot \cdot binsize \cdot texp \cdot Atel,$$

($erg2phot$) is the conversion factor between ergs and photon counts, $binsize$ is the detector bin size, $texp$ the exposure time and $Atel$ the effective telescope area. In the end the efficiency at each order is displayed, see Figure 11.2.9).

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 uves_cal_predict

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

Formatcheck frames are used to generate guess order and line tables and to automatize the order and wavelength calibration steps. They are also used to monitor the spectral format frame against predictions from the physical model and eventually detect major shifts; to measure the shift of the monitor frame against a reference frame; to measure the number of selected lines. Trending the spectral format shifts as a function of the temperature it has been possible to evidence and automatically compensate (with the instrument software) for thermal drifts in Y and X directions (see Figure 11.2.1).

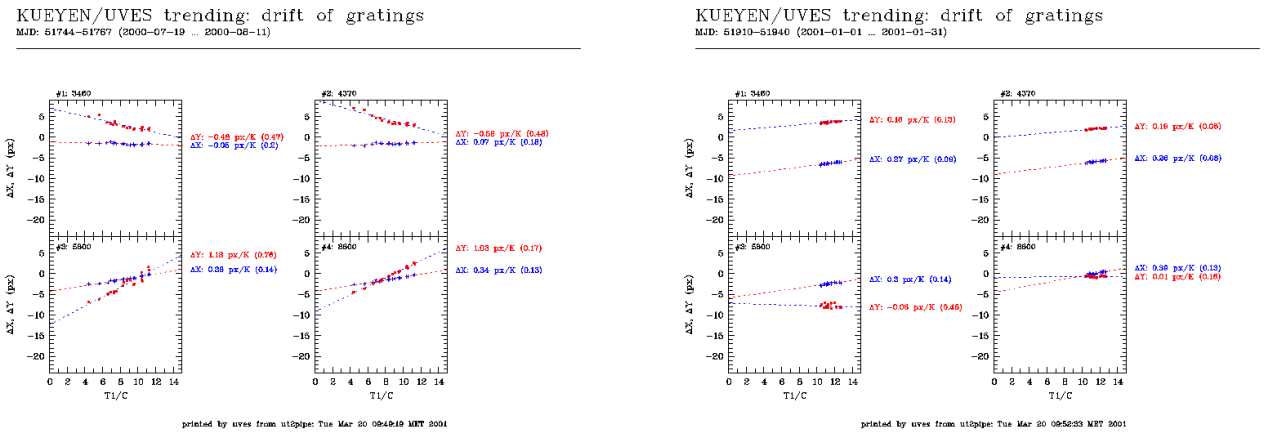


Figure 11.2.1: Thermal drift of UVES gratings before (left) and after (right) the compensation of the Y shifts introduced on Nov 2000.

As shown on figures 11.2.2 and 11.2.3, the recipe uves_cal_predict allows to monitor the spectral format against model prediction and spot if a major shift has occurred (as the case shown in the central panels of Figure 11.2.3), which may be the consequence of a major earthquake event, to eventually trigger an instrument re-alignment (to the reference position indicated by the reference formatcheck frames).

Additionally if a non well agreegated point distribution is observed in the plots shown in Figure 11.2.2, the user may decide to modify the **trans_x** and **trans_y** parameters to improove accuracy, or is warned against possible lower accuracy or failure of the following wavelength calibration data reduction step.

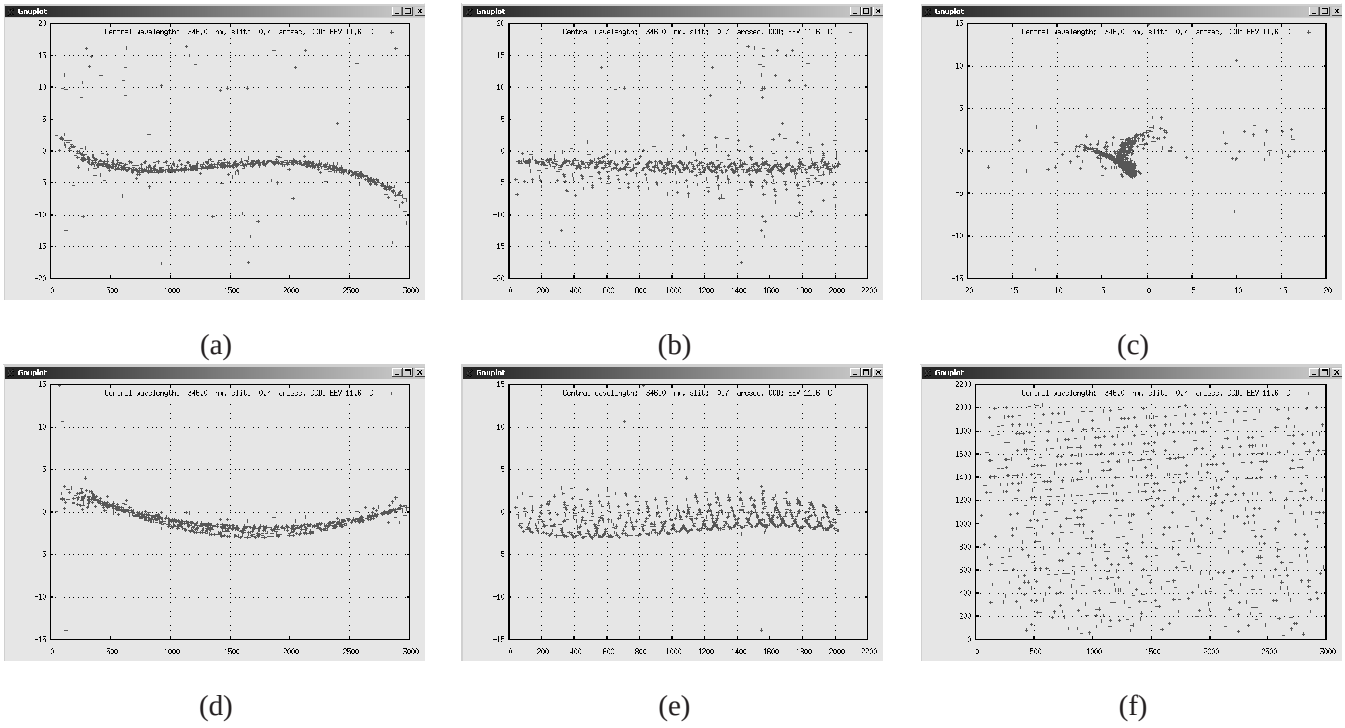
The recipes perform the following data reduction steps:

- The input formatcheck exposure is properly re-oriented (rotated, split and flipped if needed as in the RED arm case)
- Relevant ambiental and instrument setting information and the reference table lines are read to be used to compute the predicted line positions.
- Line positions are predicted and matched to the actual frame line positions.
- Guess order and line tables are built.
- Several plots are produced to assess the quality of the results.
- If a reference formatcheck frame is available a stability check is performed, by computing guess line and order tables for the current and reference formatcheck frames and by computing the correspondent shifts, performing a table cross correlation.
- The computed average and median X and Y shifts, together with the ambient temperature, pressure and instrument setting central wavelength are logged as QC parameters.

This recipe is the first step of the reduction cascade. It allows automatic data processing by finding (thanks to the use of a physical model of UVES) good initial “guess” solutions. These are necessary, for example, to obtain a stable solution in the order definition or to allow automatic wavelength calibration. Moreover it provides reference values to perform instrument quality control and stability checks. It may use a formatcheck as a reference to measure instrument spectral format shifts and eventually spot when it is necessary to realign the instrument in the event of macro Earthquakes. It is also the first recipe in a chain of data reduction recipes of the uves_obs_redchain recipe which allows automatic production of calibration solutions and full data reduction to support all of the many UVES instrument modes.

In this recipe, the geometry of the spectral format (order position and wavelength calibration) is predicted. This will be refined in successive data reduction recipes such as in the order position determination and in the wavelength calibration. Products for each chip (BLUE, REDL and REDU) are: an order table, and a line table (1st guess). This recipe generates very useful quality control plots (see Figure 11.2.2) showing respectively the shape of XDIF (difference in X between model predicted and observed line positions) vs. X (1st plot) and vs. Y (2nd plot), the YDIF (difference in Y between model predicted and observed line positions) vs. X (3rd plot) and vs. Y (4th plot), and the plots of YDIF vs. XDIF (a combined plot of the previous ones, 5th plot) and of Y vs. X (to show how each different color corresponds to a different region in the detector, 6th plot).

As described in Ballester et al., "The UVES Data Reduction Pipeline", ESO Messenger No. 101, this recipe is successful if the mentioned plots, in particular the plots showing XDIF vs X and YDIF vs Y, show a well aggregated (sigma of XDIF and YDIF $<$ 3-5 pixels) distribution of points possibly horizontally and centered at ordinate value equal to zero (see Figure 11.2.3). Vice versa, in the case of a scattered distribution of points, it means that the physical model predictions are not appropriate for the actual formatcheck frame in consideration. This may occur in the case of an instrument set-up misalignment, for example induced by a strong Earthquake, which results in formatcheck variations along Y greater than approximately 10 pixels (usually the physical model is robust enough to find a good solution for lower shifts).



(a) (b) (c)

(d) (e) (f)

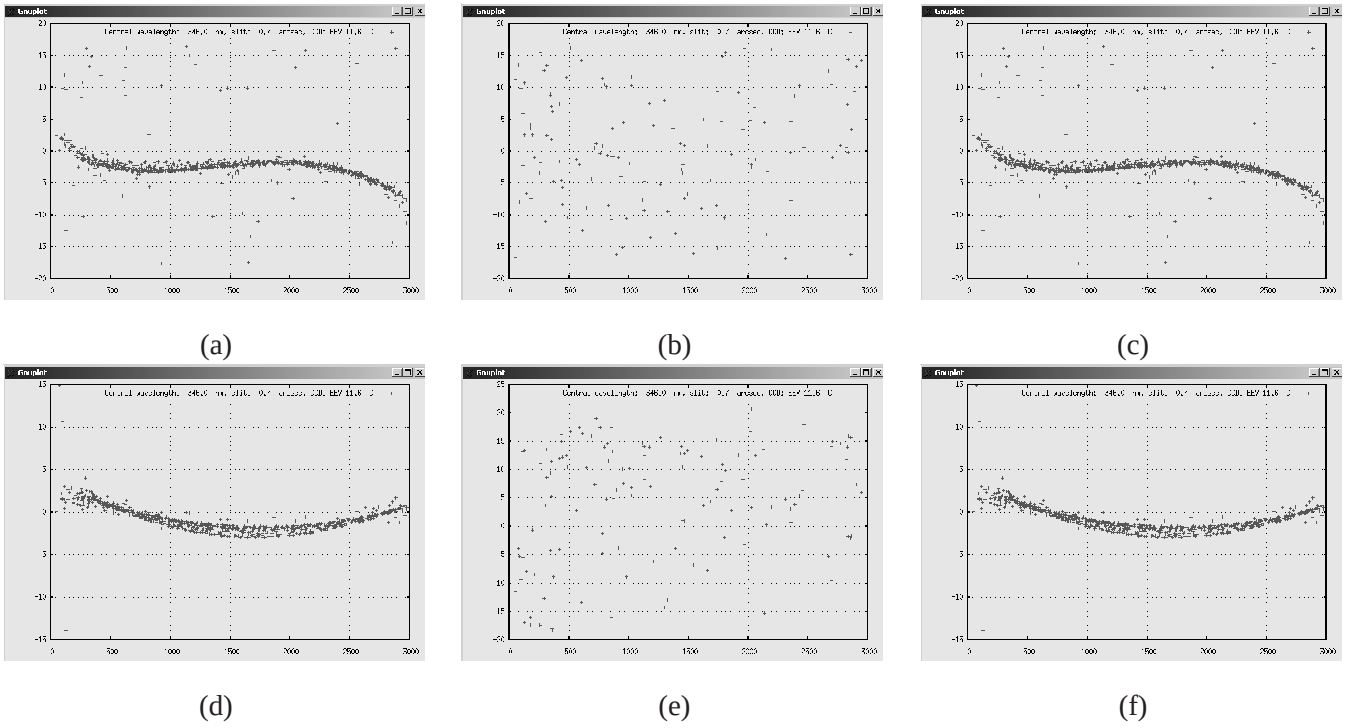
Figure 11.2.2: Physical model plots. Difference between predicted and measured X positions as a function of predicted X (a). Difference between predicted and measured Y positions as a function of predicted X (b). Difference between predicted and measured X positions as a function of predicted Y (d). Difference between predicted and measured Y positions as a function of predicted Y (e). Difference between predicted and measured Y positions as a function the difference between predicted and measured X positions (c). Detected lines positions (f). A well concentrated distribution with mean ordinate zero is an indication of good matching between the model predictions and the line positions in a formatcheck frame.

The physical model finds a good first guess solution if the actual spectral format is stable within around $\pm$ 10 pixels in X or Y. The precision of the spectral format determination as predicted by the physical model in the pipeline is the same as the one of the UVES ETC (which implements the same physical model). This precision is of the order of better than 2-3% for the Blue arm and better than 1% for the Red arm for the dispersion coordinate and better than 5 pixels on the value of the Y position. This precision refers only to the guess solution. The actual precision reached by the wavelength calibration is reported in the README file associated with the delivered data.

In case one receives data which give physical model plots with a complete scatter of points, we describe here shortly how to recover a good solution during interactive data reduction.

The parameters to be changed are (run the command **esorex --help uves_cal_predict** to get more information) **mbox_x, mbox_y, trans_x, trans_y, ech_angle_off, cd_angle_off, ccd_rot_angle_off**.

Additional recipe parameters (we recommend a non expert user not to change them) are: **compute_regression_sw**, to compute regression, **x_axis_scale**, to scale X axis, **y_axis_scale**, to scale Y axis, **def_pol1**, to set the polynomial X degree, **def_pol2**, to set the polynomial Y degree, **kappa**, to set the kappa value in kappa sigma clipping on RESIDUAL between YFIT and Y columns, **tol**, to set the tolerance in kappa sigma clipping on RESIDUAL



(a) (b) (c)

(d) (e) (f)

Figure 11.2.3: The May 12, 2000 earthquake event as detected from the physical model control plots. The normal result obtained after successful line matching (a,d) produces a well concentrated distribution with mean ordinate zero. The earthquake event causes the lines matching recipe to fail (b,e). Adjusting the model by -10 pixels (along the cross-order direction) again matches the instrument configuration (c,f).

between YFIT and Y columns.

The parameters **mbox_x, mbox_y** set the measurement box size within which a matching line is searched. This parameters should be increased to be able to more likely match the line positions as calculated by the physical model with the corresponding line positions on the detector. The detected line should fall into the box centered on the line positions calculated by the model.

The parameters **trans_x**, **trans_y** contain respectively, the X and Y translation components to be applied to the model to recover the solution. If the point distribution is scattered, it means that a significant shift (approximately $>$ 10 pixels) has occurred in one (or both) directions. In this case one should try to variate X and/or Y (one parameter at the time) by steps initially of 3 pixels until a better concentration of points is reached, refining the solution at 1 pixel step.

From pipeline version 2.1.0, in case one leaves the default setting of **trans** to 0,0, the uves_cal_predict recipe applies a shift of 6.5 pixels for the MIT (upper) chip (**trans_y=6.5**).

Other parameters to be eventually changed are **angle=ech_angle_off, ccd_angle_off, ccd_rot_angle_off**. Typically **ech_angle_off** should be kept equal to zero. More likely one should change **ccd_angle_off** and/or **ccd_rot_angle_off** (one at the time) in recipes of 0.01-0.1 deg until a better distribution is reached.

The parameters **trans_x, trans_y** and **ech_angle_off, ccd_angle_off, ccd_rot_off** should be varied one at a time initially chosing a high value of **mbox_x, mbox_y** (80, 80 or 70, 70) and next, when a good solution is found,

verifying that the quality of the plot remains almost the same even when one decreases the value of **mbox_x, mbox_y** parameters to 40, 40 (standard values). Shifts of instrument spectral format are more frequent along the Y direction, which could be recovered by modifying just the corresponding **ech_angle_off, ccd_angle_off, ccd_rot_off** parameter. In case of non-standard observational settings taken with a camera tilt (which implies in an **ech_angle_off, ccd_angle_off, ccd_rot_off** along X direction), if any data reduction problem is noticed, one can try to recover the solution by adjusting only the offset along the X direction.

To proceed easily and quickly in this operation one should follow the following steps:

1. Use as master formatchecks, frames that result in physical model plots with a very good aggregation, and having the same instrumental setting as the one which is not giving a very precise aggregation, and set their ESO.PRO.CATG value to MASTER_FORM_x (x= BLUE or REDL, REDU). These can be found in directory fluves/calib of the release tar file.

2. Use them and the current formatcheck frame to quickly evaluate Y and X **trans_x, trans_y** (shifts), by comparing the images.

3. Use the master formatchecks (lower and upper chip frames) and the Th-Ar Line reference table, put them in a reference catalogue, and use the recipe uves_cal_predict to do the formatcheck recipe on the formatcheck data, using as **mbox_x** and **mbox_y** (match window) parameter values 80,80, and use as **trans_x, trans_y**, the values previously determined.

4. Iterate the previous step by adjusting (**trans_x, trans_y**) and using the information provided by the X shift and Y shift values reported from the stability test until the points converge. In practice one has to add the value (with sign) of the found shift to the corresponding **trans_x, trans_y** component until the value of the shifts become less than one pixel for each component.

5. Try to reduce the (**mbox_x, mbox_y**) values gradually to 40, 40. Repeat the two previous steps. When the X and Y shifts are smaller in absolute value than 5 pixels start to reduce the (**mbox_x, mbox_y**) values at 10 pixels steps from 80, 80 to 40, 40 until one get with 40, 40 X and Y shifts values less than one pixel shift.

11.2.2 uves_cal_orderpos

This recipe measures the echelle order positions.

The inputs are a short slit flat field frame and optionally, to ensure robustness in the orders' detection, a guess order table.

The product is an order table.

- A median filter of radii **radx**, **rady** is applied to the input frame.
- The interorder background is subtracted using polynomial method (see section 11.1.2).
- If no guess order table is provided, the orders are detected by means of a Hough transformation (see 11.1.3).

- The orders are then traced (see 11.1.4), and the positions are finally fitted with a global polynomial of automatic degree. The polynomial degrees are increased in steps of one or two, as long as the fit RMS decreases by more than 10%.

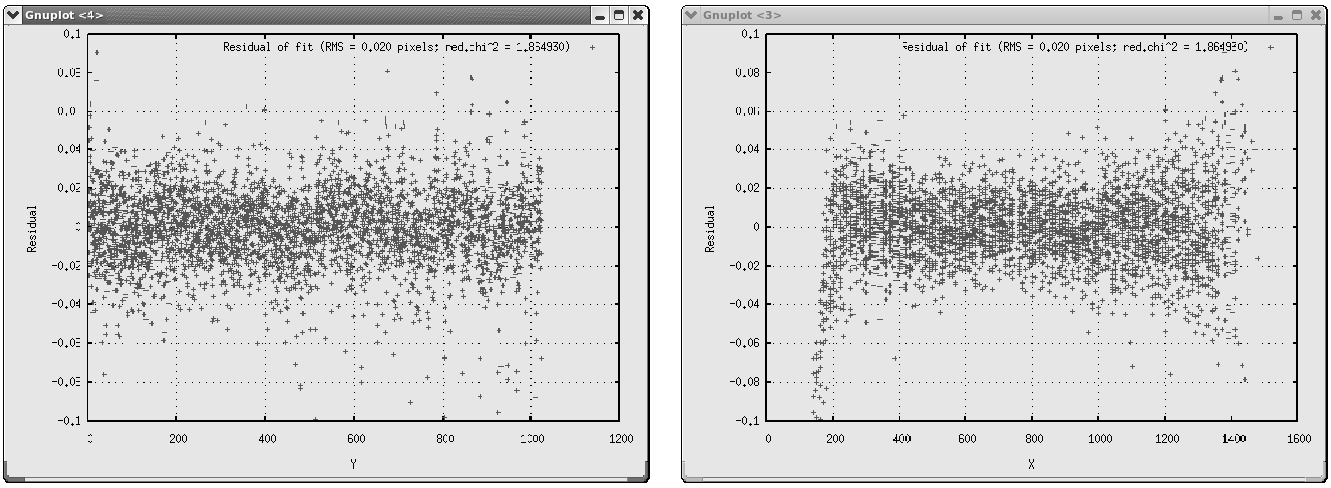


Figure 11.2.4: Order definition residual plots as a function of X (left) and of the detected order (right)

11.2.3 uves_cal_mbias

This recipe generates a master bias frame from a set of biases.

The master frame is created by median stacking.

The detector read out noise (RON) is measured in a central box 100 $\times$ 100 pixels wide, both on the raw and the master frame, to measure the efficiency of the master creation process. The RON should decrease as the root mean square of the number of coadded frames (typically 5), which is monitored by the FITS keyword PRO.DATANCOM.

The parameters QC.OUT1.STRUTCTX and QC.OUT1.STRUCTY is a way to monitor more accurately the stability of a frame. It is obtained by collapsing the frame along a given direction. The corresponding information can be displayed on a 2D plot to spot local frame dis-omogeneity or changes with the time. The values QC.OUT1.STRUCTX and QC.OUT1.STRUCTY are the clean RMS computed on the frame collapsed respectively along X or Y. A sudden change of those parameters may indicate a local change of the master frame behaviour.

Additionally, trending the value of PRO.DATAMED measured on the master frame, it is possible to monitor the median of the bias level to verify the detector health. Examples of trending plots are on Figure 11.2.5.

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).

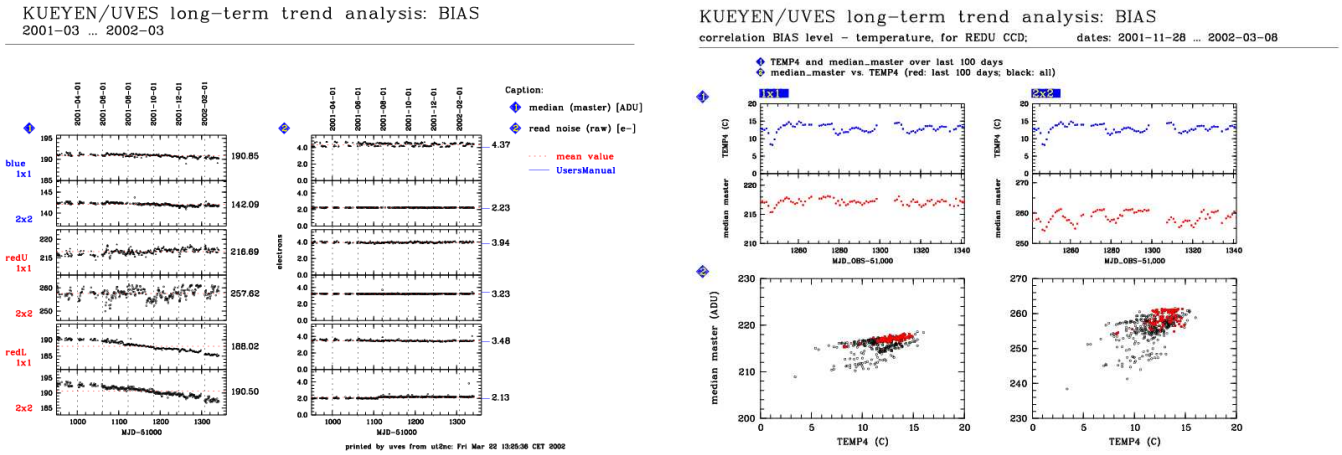


Figure 11.2.5: Trending plot of the median value for a master bias and the noise computed on a raw bias for different instrument settings (left panel). The REDU CCD shows pronounced fluctuations around a stable mean value which are at least partly correlated with temperature. In the trending plot (for 1x1 and 2x2 binnings) shown in the right panel, the first row shows the temperature(TEMP4), the second row the median BIAS level. Both are plotted within a window of 100 days (first and last date given in the header). The third row shows the correlation of both parameters. While the unbinned (1x1) BIASes follow nicely the temperature variations, the binned BIAS frames show superimposed on similar variations also apparently random variations on a timescale of a few days.

The master frame is created by median stacking, and the master bias is subtracted.

11.2.5 uves_cal_mflat

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

The master frame is created by median stacking, dark subtraction (if provided) and bias subtraction. The master flat is also background subtracted using the spline method (see section 11.1.2).

11.2.6 uves_cal_wavecal

This recipe performs the wavelength calibration using previously determined solutions for the first guess line table and the order table. By default the slit is divided into three subslits of equal length, and a wavelength calibration solution is produced for each of the subslits.

The steps performed during the calibration of a given extraction window are the following:

- If a master bias is provided, it is subtracted from the arc lamp image.
- If a master flat is provided, the arc lamp image (previously bias corrected) is divided by the flat-field. This is a recommended step because the flat-field division systematically shifts the positions of the ThAr lines

(due to the superimposed blaze function). If the arc lamp frame is not flat-fielded the obtained dispersion solution would be incompatible with the science frame which is usually flat-fielded.

- The spectrum is extracted using average, linear or object-weighted extraction (See 11.1.10, 10.7.5).
- Emission line detection in the extracted spectrum. (See 11.1.5).
- Computation of the absolute (physical) numbering of the orders using the provided polynomial map from (pixel, pixel)-space to physical order numbers.
- Obtain a first solution by estimating the **x-shift** of the detected lines with respect to the locations of the lines in the provided "guess" line table, then apply this shift to the "guess" dispersion relation. (See 11.1.6).
- Iterative line identification and update of the dispersion relation, until no new identifications can be made. (See 11.1.7).

For quality control purposes, it is important to check the plot of the residual difference in wavelength between the lines listed in the reference line table and the corresponding measured line positions. This plot gives an indication of the accuracy (the ordinate is expressed in Angstrom units) of the wavelength calibration as function of order number (see Figure 11.2.6).

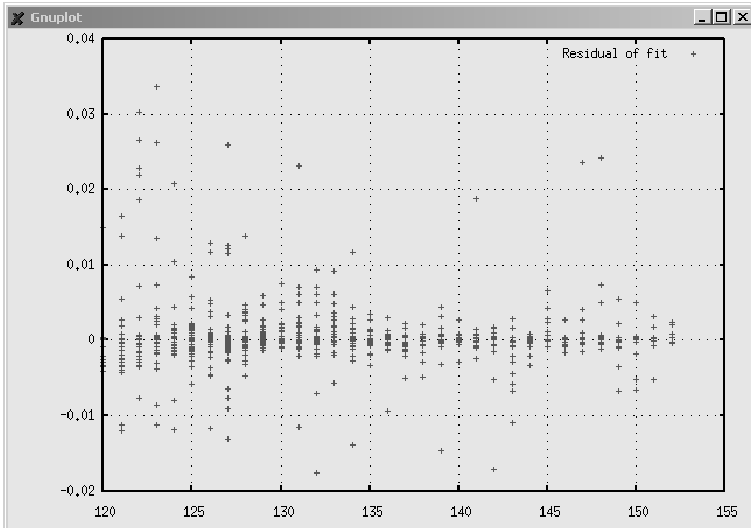
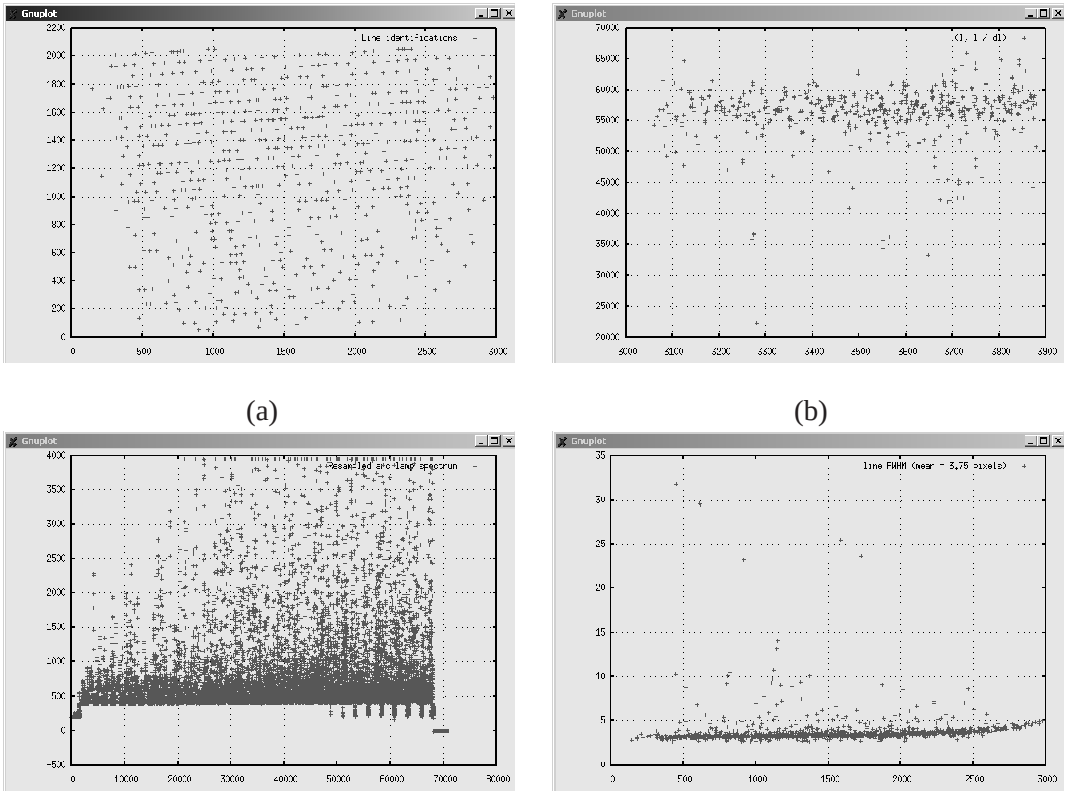


Figure 11.2.6: Wavelength calibration residuals (A) as function of order number.

After having determined the wavelength calibration solution for the object extraction window, resolution plots are produced.

These plots display the distribution of identified lines across the detector, the resampled ThAr spectrum, the resolving power as a function of wavelength, the line FWHM as a function of wavelength.

The obtained wavelength solution is applied during the science reduction when the spectrum is resampled from pixels to equidistant wavelength bins. Due to the non-zero line tilt (see figure 11.2.8) the wavelength solutions



(a) (b)

(c) (d)

Figure 11.2.7: These plots display: (a) the distribution of identified lines across the detector, (b) the resolving power as a function of wavelength, (c) the resampled ThAr spectrum, (d) the line FWHM as a function of wavelength.

obtained at different slit offsets are interpolated (using linear interpolation) to the measured object offset position). In the case of object-weighted arclamp extraction (see 10.7.5) the line table contains three identical solutions, and the linear interpolation has no effect.

11.2.7 uves_cal_response

This recipe determines the response function and quantum efficiency.

During this recipe the instrument response function and the telescope+instrument+detector detection efficiency are determined. First the standard star spectrum is optimally extracted (see 11.1.8).

Then the response of the instrument is determined by comparing the wavelength calibrated frame with the corresponding reference spectrum of a catalog.

The reference spectrum is then extracted from the table by comparing the coordinates of the standard star measurement with the ones of the reference standard star catalog and storing the resulting spectrum in a table.

Next, the instrument response is determined, running first a median filter on the reduced input spectrum, and

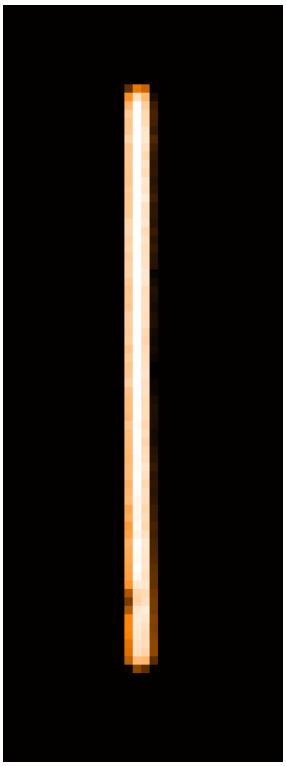


Figure 11.2.8: The cross-dispersion direction is not perfectly aligned with the detector columns. The centroid position of the ThAr emission line shown here (from lower, red chip, central wavelength = 860nm) changes up to around 0.5 pixels as function of slit position. Since the wavelength calibration RMS is usually less than 0.1 pixels it is important to resample the science object using a dispersion solution that corresponds to the science object offset.

a smooth filter, to smooth out sudden jumps. Next the ratio between the reference flux table and the extracted standard star spectrum is calculated. This ratio is cleaned to eliminate the possibility of having boundary effects. This finally gives the instrument response.

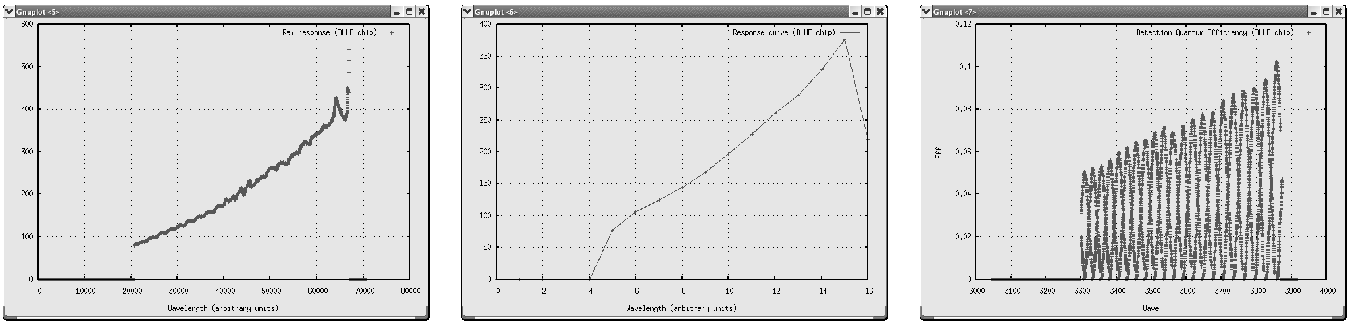
The main data product of this recipe is the merged response. This is used to flux calibrate the reduced merged science spectrum (see Figure 11.2.9 panels (a) and (b)). Other products are the wavelength calibrated response and the reference flux table of the standard star.

Then is computed the efficiency of the telescope+instrument+detector (see Figure 11.2.9 panel (c) and section 11.1.13).

11.2.8 uves_obs_scired

This recipe reduces a science frame by using a combination (depending on recipe parameters and provided input frames) of the steps:

- The master bias is subtracted,
- The master dark (if provided) is subtracted,
- The interorder background is estimated as described in 11.1.2 and then subtracted,
- If **reduce.ffmethod** is set to “pixel” the corrected object image is divided by the master flat field.



(a) (b) (c)

Figure 11.2.9: The UVES pipeline also produces instrument response curves (raw (a) and smoothed (b)) and plots of the instrument efficiency (c). The last could be used to monitor the instrument health and, from time to time, upgrade the UVES Exposure Time Calculator predictions.

- Then the object is extracted using either the optimal (11.1.8, the linear (11.1.9), the average (11.1.10) or the 2D (11.1.11) method. If the spectrum is optimally extracted, cosmic rays are detected and rejected, and the sky is subtracted during the extraction. If linear/average extraction is used, the sky is extracted in windows above and below the object and subsequently subtracted.
- If **reduce.ffmethod** is set to “extract”, the master flat field is extracted using the previsouly obtained object weights, and the object spectrum is divided by the extracted master flat field spectrum.
- The spectrum is resampled to equidistant wavelength steps with size defined by the parameter **reduce.rebin.wavestep**, and optionally (depending on **reduce.rebin.scale**) multiplied by $\frac{d\lambda}{dx}$ to ensure flux conservation.
- Then if **reduce.skysub** is set to TRUE the sky is subtracted.
- The orders are merged using one of the two possible methods set by the parameter **reduce.merge**.
- Finally, the spectrum is moved to the top of the atmosphere (by correcting for atmospheric extinction if the atmospheric dispersion table is provided) and flux calibrated to physical units (if the instrument response curve is provided).

11.2.9 uves_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)
- uves_cal_predict (if no guess order and line tables are provided)
- uves_cal_orderpos (if no order table is provided)

- uves_cal_mflat (if no master flat is provided)
- uves_cal_wavecal (if no dispersion solution line table is provided)
- uves_cal_response (if a standard star is provided)
- uves_obs_scired (unless the option **scired** is set to FALSE)

11.2.10 uves_cal_cd_align

This is a special recipe used operationally to monitor the stability and reproducibility of the cross disperser position. The recipe measures the shift between two frames each containing one echelle order.

11.2.11 uves_cal_tflat

This is a special recipe used operationally to monitor the stability of the instrument (see also 7.6). It reduces a telluric line flat field frames: it generates a MASTER_TFLAT_x frame, divide the first raw TFLAT frame by a reference TFLAT frame, reduces the first raw TFLAT frame as a science frame using as flat field the MASTER_TFLAT_x frame previously generated.

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,
- 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 or newer)

A.3 Building the UVES pipeline

The UVES pipeline distribution kit contains:

uves-manual-12.0.pdf	The UVES pipeline manual
install_pipeline	Install script
qfits-6.2.0.tar.gz	QFITS 6.2.0
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-4.3.0.tar.gz	UVES 4.3.0
uves-calib-4.3.0.tar.gz	UVES static calibration files 4.3.0

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the UVES pipeline 4.3.0 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-4.3.0.tar.gz**

4. Unpack using the following commands:
   gunzip uves-kit-4.3.0.tar.gz tar -xvf uves-kit-4.3.0.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-4.3.0

   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 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