

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 6.6.1

Date 2026-05-05

Prepared:	ESO UVES Pipeline Team	2026-05-05	
	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
13.0	03/11/2009		Update to 4.4.8
14.0	30/06/2010		Update to 4.7.4
15.0	10/08/2010		Support for UVES Reflex BLUE arm workflow
16.0	28/01/2011		Support for UVES Reflex RED and BLUE arms workflow
17.0	15/04/2011		Updated to 4.9.5
18.0	31/10/2011		Updated to 4.9.8
19.0	05/04/2012		Reflex2 support
20.0	15/05/2012		Updated to 5.0.7
21.0	22/10/2012		Updated to 5.0.9
22.1	15/10/2012		Updated to 5.0.17
22.2	21/12/2012		Updated to 5.1.0
22.3	06/05/2013		Updated to 5.2.0
22.4	09/09/2013		Updated to 5.2.5
22.5	21/10/2013		Updated to 5.3.0
22.7	25/03/2014		Updated to 5.4.0
22.8	05/11/2014		Updated to 5.4.3
22.9	18/03/2015		Updated to 5.5.2
22.10	20/07/2015		Updated to 5.5.5
22.11	14/09/2015		Updated to 5.5.7
22.12	01/03/2016		Updated to 5.7.0
22.13	03/03/2016		Updated to 5.7.1
22.15	19/10/2017		Updated to 5.8.3
6.1.3	03/06/2020		Updated to 6.1.3
6.1.6	20/05/2021		Updated to 6.1.6
6.1.7	15/03/2022		Updated to 6.1.7
6.1.8	19/05/2022		Updated to 6.1.8
6.6.1	05/05/2026		Updated to 6.6.1

This page was intentionally left blank

Contents

1 Introduction **11**

1.1 Purpose . . . 11

1.2 Acknowledgements . . . 11

1.3 Scope . . . 11

1.4 Conventions . . . 12

1.5 Reference documents . . . 12

2 Overview **15**

3 What's new in pipeline release 6.6.1 **16**

4 UVES Instrument Description **17**

4.1 Instrument overview . . . 17

5 Quick start **19**

5.1 UVES pipeline recipes . . . 19

5.2 An introduction to Reflex, Gasgano and EsoRex . . . 20

5.2.1 Using *Reflex* . . . 20

5.2.2 Example of data reduction using the Reflex-based UVES workflow . . . 21

5.2.3 Using *Gasgano* . . . 26

5.2.4 Using EsoRex . . . 30

5.2.5 Example of data reduction using EsoRex . . . 33

5.3 FITS header keywords for raw data identification . . . 34

6 Known problems **40**

7 Instrument Data Description **42**

7.1 Format check frames . . . 42

7.2 Bias frames . . . 42

7.3 Dark frames . . . 42

7.4 Order definition frames . . . 42

7.5 Flat field frames . . . 42

7.6 Telluric lamp flat field frames . . . 43
7.7 Arc frames for wavelength calibration . . . 43
7.8 Standard star frames . . . 43
7.9 Supported raw frames (keyword identifiers) . . . 43

8 Static Calibration Data 52
8.1 Emission line reference table . . . 52
8.2 Standard stars flux table . . . 52
8.3 Atmospheric extinction table . . . 52
8.4 Table to monitor line intensity . . . 52
8.5 Master response curves . . . 53
8.6 Table with wavelength points to fit the response . . . 53

9 Data Reduction 54
9.1 Data reduction overview . . . 54
9.2 Required input data . . . 54
9.3 Reduction cascade . . . 55

10 Pipeline Recipe Interfaces 57
10.1 uves_cal_mbias . . . 58
10.1.1 Input . . . 58
10.1.2 Output . . . 58
10.1.3 Quality control . . . 58
10.1.4 Parameters . . . 59
10.2 uves_cal_mdark . . . 59
10.2.1 Input . . . 59
10.2.2 Output . . . 59
10.2.3 Quality control . . . 60
10.2.4 Parameters . . . 60
10.3 uves_cal_predict . . . 61
10.3.1 Input . . . 61
10.3.2 Output . . . 61

10.3.3 Quality control . . . 62
10.3.4 Parameters . . . 63
10.4 uves_cal_orderpos . . . 63
10.4.1 Input . . . 63
10.4.2 Output . . . 63
10.4.3 Quality control . . . 64
10.4.4 Parameters . . . 64
10.5 uves_cal_mflat . . . 65
10.5.1 Input . . . 65
10.5.2 Output . . . 65
10.5.3 Quality control . . . 65
10.5.4 Parameters . . . 66
10.6 uves_cal_mflat_combine . . . 66
10.6.1 Input . . . 66
10.6.2 Output . . . 67
10.6.3 Quality control . . . 67
10.6.4 Parameters . . . 67
10.7 uves_cal_mkmaster . . . 67
10.8 uves_cal_wavecal . . . 67
10.8.1 Input . . . 67
10.8.2 Output . . . 68
10.8.3 Quality control . . . 70
10.8.4 Parameters . . . 70
10.8.5 Object-weighted extraction . . . 71
10.9 uves_cal_response . . . 72
10.9.1 Input . . . 72
10.9.2 Output . . . 72
10.9.3 Quality control . . . 73
10.9.4 Parameters . . . 73
10.10uves_obs_scired . . . 74

10.10.1 Input . . . 74
10.10.2 Output . . . 75
10.10.3 Quality control . . . 75
10.10.4 Parameters . . . 77
10.11 uves_obs_redchain . . . 78
10.11.1 Input . . . 78
10.11.2 Output . . . 79
10.11.3 Quality control . . . 79
10.11.4 Parameters . . . 79

11 Algorithms and recipe details **80**
11.1 Algorithms . . . 80
11.1.1 Error model . . . 80
11.1.2 Master frame creation using a kappa-sigma-clipped mean . . . 80
11.1.3 Guess table generation . . . 80
11.1.4 Reference arc line position measure . . . 81
11.1.5 Guess line and order tables generation . . . 81
11.1.6 Spectral format stability check . . . 82
11.1.7 Combination of blue master ^{2}D and normal flat frames . . . 82
11.1.8 Interorder background subtraction . . . 82
11.1.9 Hough transform . . . 83
11.1.10 Order tracing . . . 83
11.1.11 Line search . . . 84
11.1.12 Determination of the wavelength calibration . . . 84
11.1.13 Optimal extraction . . . 85
11.1.14 Linear extraction . . . 89
11.1.15 Average extraction . . . 89
11.1.16 Spatial extraction. . . . 90
11.1.17 Reduction of more than one object on the slit . . . 90
11.1.18 Reduction of observations acquired using the image slicer . . . 90
11.1.19 Response computation . . . 91

11.1.20 Flux calibration . . . 91
11.1.21 Efficiency computation . . . 91
11.1.22 Spectrum flux calibration . . . 92
11.2 Recipes . . . 92
11.2.1 uves_cal_mbias . . . 92
11.2.2 uves_cal_mdark . . . 95
11.2.3 uves_cal_predict . . . 95
11.2.4 uves_cal_orderpos . . . 99
11.2.5 uves_cal_mflat . . . 100
11.2.6 uves_cal_wavecal . . . 100
11.2.7 uves_cal_response . . . 103
11.2.8 uves_obs_scired . . . 104
11.2.9 uves_obs_redchain . . . 106
11.2.10 uves_cal_cd_align . . . 107
11.2.11 uves_cal_tflat . . . 107

A Workflow technicalities **108**
A.1 Data requirements and optional inputs . . . 108
A.2 Overall layout . . . 109
A.3 OCA rules . . . 109
A.4 Workflow fine tuning and hints . . . 110

B Installation **112**
B.1 Supported platforms . . . 112
B.2 Requirements . . . 112
B.3 Building the UVES pipeline . . . 113
B.3.1 Installing the UVES data demo package . . . 114

C Abbreviations and acronyms **116**

1 Introduction

1.1 Purpose

The UVES pipeline is a subsystem of the *VLT Data Flow System* (DFS). Its target user is ESO *Quality Control and Data Product* (QCDP) 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 May 5, 2026 (version 6.6.1). Release 6.6.1 supports the reduction of UVES frames obtained in the UVES standard modes. [1]

1.2 Acknowledgements

The UVES pipeline was 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 for the data reduction of data proceding 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 was 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, Daniel Bramich.

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 for 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 [13]. For general information about the current instrument pipeline status we remind the user of [5]. Quality Control information are at [4].

Additional information on the Common Pipeline Library (CPL) and EsoRex can be found respectively at [9], [11]. The Reflex front end is described in [22]. The Gasgano front end is described in [12]. Description of the instrument are in [6], [16],[8]. The UVES instrument user manual is in [8]. The UVES calibration plan is in [3], while results of Science Verifications (SV) are at [7]. Additional information on the DFS and VLT data interfaces are in [2], [10] and [18]. A clear and compact description of the UVES pipeline is in [15].

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

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

1.4 Conventions

Throughout this document, we use several conventions to help simplify the text:

- For generic filenames that refer to an unspecified *arm* of the UVES instrument, the `_<arm>` suffix may indicate either `_blue` or `_red` (example: `/path_raw/uves_bias_<arm>.fits`)
- For generic filenames that refer to an unspecified *detector chip* in the UVES instrument, the `_<chipid>` suffix may indicate either `_blue`, `_redl`, or `_redu` (example: `masterdark_<chipid>.fits`)
- For generic `PRO.CATG` TAGs that refer to an unspecified *arm*, the `_<ARM>` suffix may indicate either `_BLUE` or `_RED` (example: `ARC_LAMP_<ARM>`)
- For generic `PRO.CATG` TAGs that refer to an unspecified *detector chip*, the `_<CHIPID>` suffix may indicate either `_BLUE`, `_REDL`, or `_REDU` (example: `MASTER_BIAS_<CHIPID>`)

1.5 Reference documents

[1] P. Ballester. *Hough Transform for robust regression and automated detection*. ESO/SDD/PSD, ADS. A&A, **286**,1011, 1994. 83

[2] ESO. *VLT Data Flow System Specifications for Pipeline and Quality Control*. VLT-SPE-ESO-19600-1233. 12

[3] ESO, . *VLT UVES Calibration Plan*. VLT-PLA-ESO-13200-1567. 12

[4] ESO/DMO/DFO, *http://www.eso.org/observing/dfo/quality/*. *ESO-Data Flow Operation home page*. 12

[5] ESO/DMO/DFO, *http://www.eso.org/sci/software/pipelines/*. *UVES Pipeline Current Status*. 12

[6] ESO/INS, *http://www.eso.org/sci/facilities/paranal/instruments/uves*. *UVES instrument home page*. 12, 44

[7] ESO/INS, *http://www.eso.org/sci/activities/vltsv/fors2-uves/UVES_SV.html*. *UVES SV home page*. 12

[8] ESO/INS, *http://www.eso.org/sci/facilities/paranal/instruments/uves/doc*. *UVES User Manual*. VLT-MAN-ESO-13200-1825. 12, 17

[9] ESO/SDD/DFS, *http://www.eso.org/cpl/*. *CPL home page*. 12

[10] ESO/SDD/DFS. *DFS Pipeline & Quality Control – User Manual*. VLT-MAN-ESO-19500-1619. 12

[11] ESO/SDD/DFS, *http://www.eso.org/cpl/esorex.html*. *ESOREX home page*. 12, 15, 31, 32

[12] ESO/SDD/DFS, *http://www.eso.org/gasgano/*. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 12, 15, 28

[13] ESO/SDD/DFS, *http://www.eso.org/pipelines*. *UVES Pipeline User Manual*. VLT-MAN-ESO-19500-2964. 12

[14] ESO/SDD/SPD, *https://ftp.eso.org/pub/dfs/pipelines/instruments/uves/uves-reflex-tutorial-6.6.1.pdf*. *Reflex UVES Tutorial*, 2012. VLT-MAN-ESO-19500-5296. 20, 21

[15] P. Ballester et al. *The UVES Data Reduction Pipeline*. ESO/SDD/PSD, *http://www.eso.org/sci/publications/messenger/archive/no.101-sep00/messenger-no101-31-36.pdf*. The Messenger, 31-36, **101**. 12

[16] S. D'Odorico et al. Successful Commissioning of UVES at Kueyen. ESO/SDD/PSD, *http://www.eso.org/sci/publications/messenger*. The Messenger, **99**, 1, 2000. 12, 17

[17] K. Horne. *An optimal extraction algorithm for CCD spectroscopy*. ADS. PASP, **98**,609, 1986. 85

[18] *http://archive.eso.org/cms/tools-documentation/eso-data-interface-control.html*. *ESO DICB – Data Interface Control Document*. ESO/DMO. GEN-SPE-ESO-19400-0794 (5.0). 12, 44

[19] Kepler Project, code.kepler-project.org/code/kepler-docs/trunk/outreach/documentation/shipping/getting-started-guide.pdf. *Kepler Getting Started Guide*. 20

[20] Marsh. *The extraction of highly distorted spectra*. ADS. PASP, **101**,1032, 1989. 85, 86

[21] Mukai. *Optimal extraction of cross-dispersed spectra*. ADS. PASP, **102**, 183, 1990. 85

[22] Reflex Project, *http://www.eso.org/sci/software/reflex/*. *Reflex*. 12, 15

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, from *Reflex* or from *Gasgano*.

ESO offers three front-end applications for launching pipeline recipes; namely *Reflex* [22], *Gasgano* [12] and *EsoRex* [11]. The last two are included in the pipeline distribution (see Appendix B, page 112). These applications can also be downloaded separately from `www.eso.org/reflex`, `www.eso.org/gasgano`, `www.eso.org/cpl/esorex.html` respectively.

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 Reflex or 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 are 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 B, the installation of the UVES pipeline recipes is described, and in Appendix C, a list of used abbreviations and acronyms is given.

3 What's new in pipeline release 6.6.1

The following major changes have been implemented in the UVES pipeline (with respect to the previous CPL based pipeline):

- compute several additional QC parameters directly within the pipeline
- barycentric correction of spectrographic data
- updated to latest version of CPL (v7.3)

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 [8].

4.1 Instrument overview

UVES is a two-arm, cross-dispersed echelle spectrograph covering the wavelength range 300 - 500 nm (blue) and 420 - 1100 nm (red) with the possibility of employing dichroics (see Figures 4.1 and 4.2). 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 instrument refer to e.g. [16]. 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 Nasmyth A platform of UT2 of the VLT. The fibre link to the UVES red arm is fed by eight fibres with a nominal resolving power of R=47000.

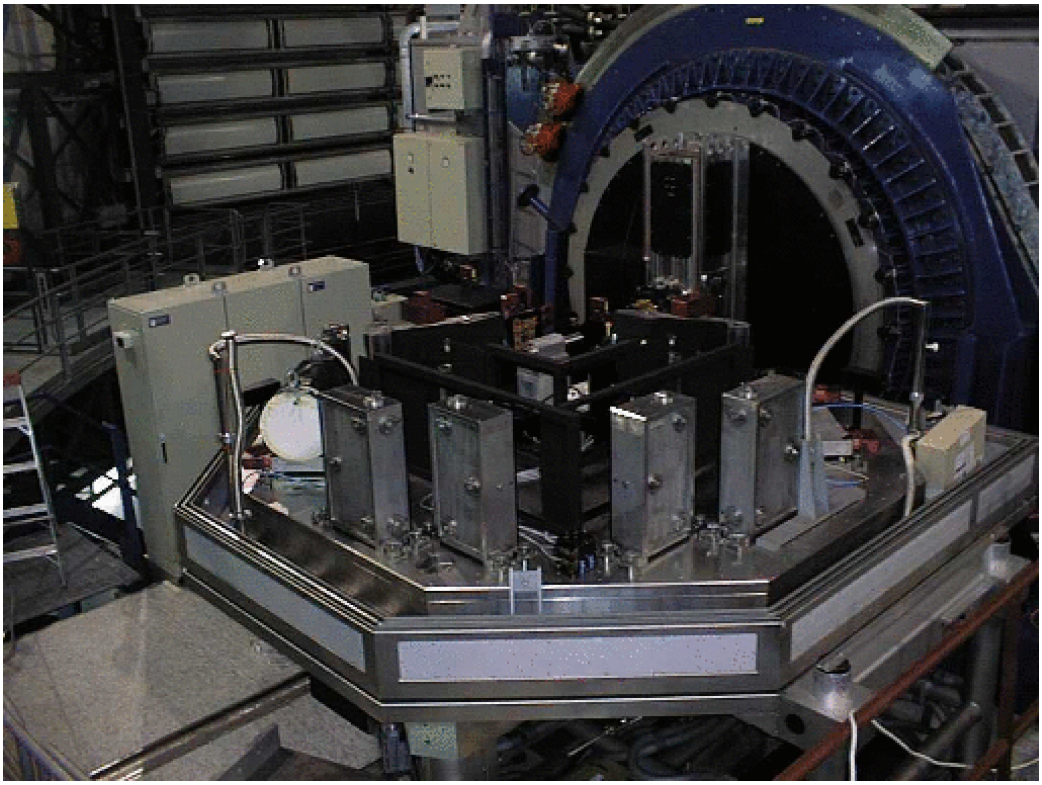


Figure 4.1: A photo of UVES mounted at the Nasmyth B focus of Kuyen (VLT-UT2).

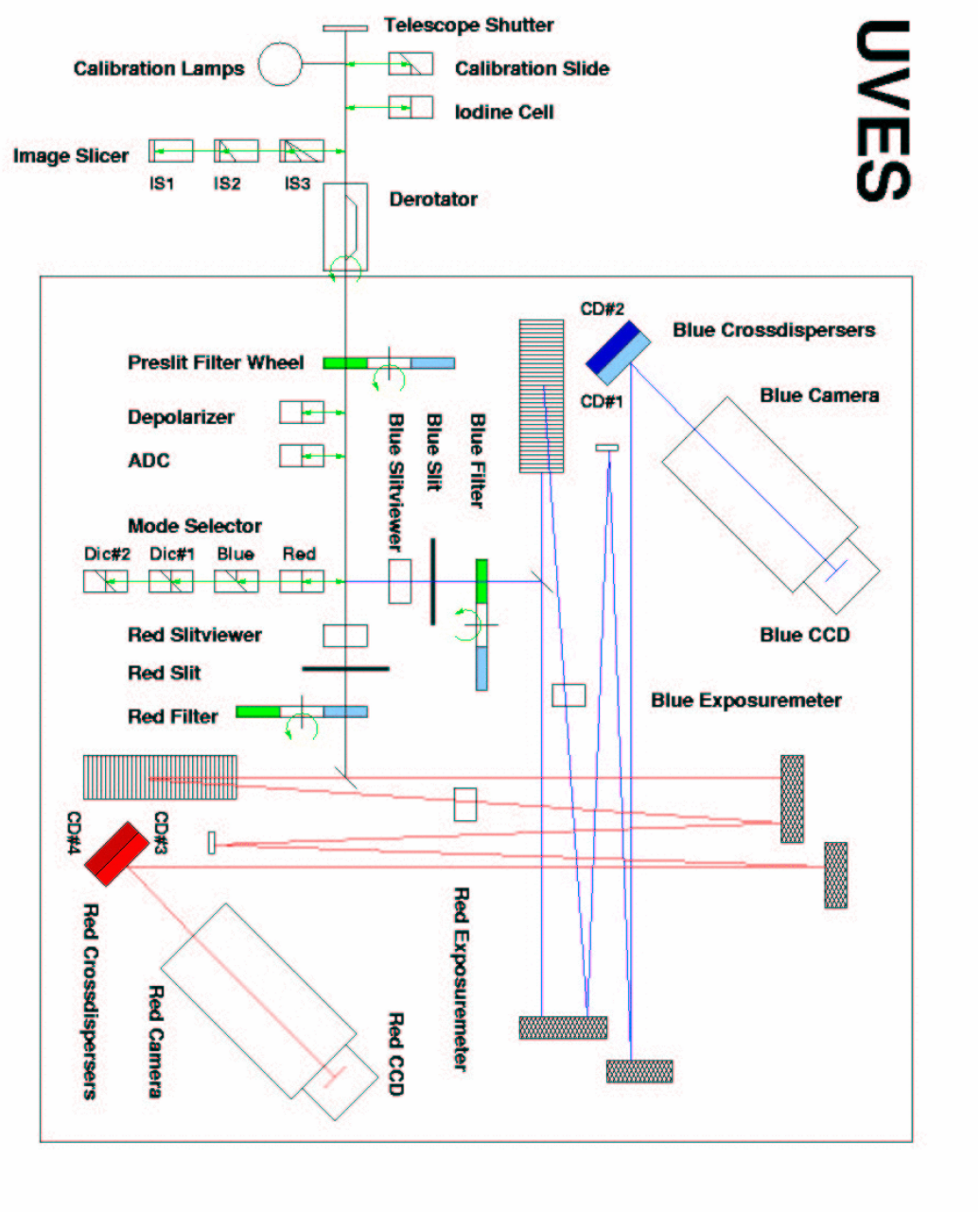


Figure 4.2: A schematic overview of the UVES spectrograph.

5 Quick start

This section describes how to make 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 the 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, without checking the results of each stage, the user may employ the **uves_obs_redchain** recipe.

Additionally, the pipeline provides four further recipes:

uves_cal_mflat_combine: It can be used to combine a `MASTER_DFLAT_BLUE` with a `MASTER_FLAT_BLUE` in a composite `MASTER_FLAT_BLUE`. Using this last frame during science data reduction extraction improves quality.

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: Used operationally to measure the reproducibility of the cross disperser position.

5.2 An introduction to Reflex, Gasgano and EsoRex

Before being able to call pipeline recipes to process a set of data, the data must be correctly 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 observing 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. Since all the required information is stored in the FITS headers, data association is based on a set of header keywords (called "association keywords") and the process 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, namely:

- *Reflex* is a graphical tool that helps the user to execute data reduction workflows which contain several recipes. This dramatically decreases the time the user needs to run a whole reduction chain, from calibration and raw data down to the final products. *Reflex* takes care of grouping the different data sets, associating the calibration frames and managing the interdependencies between recipes in the calibration cascade. **Reflex is the recommended software tool for reducing your data.**
- *Gasgano* is an alternative data management tool that simplifies the data organization process. In addition, *Gasgano* allows the user to execute directly the pipeline recipes on a set of selected files.
- *EsoRex* is a command line tool used to run the pipeline recipes. *EsoRex* commands can be easily scripted.
- The Paranal observatory implements automatic data management tools that trigger the execution of pipeline recipes. This aspect is not covered in this manual.

5.2.1 Using *Reflex*

Reflex is the recommended tool to reduce complete data sets that include all the calibration frames. It is an advanced tool, and yet easy to use, that is geared towards maximum scientific return. It is based on the workflow engine *Kepler* [19].

This manual does not cover the installation of *Reflex* . Please refer to [14] for the installation procedure which also contains a detailed description of the *Reflex* application. What follows is a very brief summary of it.

Once installed, *Reflex* can be executed with the command:

```
user@host# reflex &
```

Reflex main concepts are workflows and actors. Workflows are canvasses which show the interdependence of the pipeline recipes, allowing the user to easily obtain an overview of the reduction steps. Workflows have the advantage of requiring a small learning curve in order to get the pipeline running.

Actors are the entities which actually perform some kind of operation. In *Reflex*, to each main actors correspond the pipeline recipes themselves, which perform the data reduction steps, but there are other actors such as the

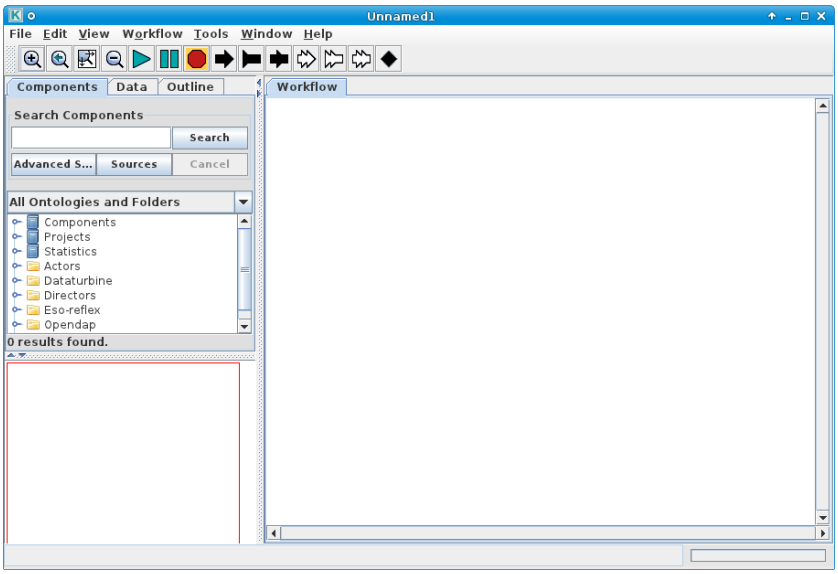


Figure 5.1: *Fresh Reflex canvas.*

DataOrganizer, or the FitsRouter that are useful to manage the data files. Each actor can be configured by right-clicking on it and selecting *Configure Actor* as shown in Figure 5.2. In the case of the recipe actors, the recipe parameters are part of the actor and make up the second group of parameters.

In addition to those elements, the workflow contains variables that contain the most important settings, such as the directories where data is located and will be saved.

To start using *Reflex* with this pipeline, please refer to [14].

5.2.2 Example of data reduction using the Reflex-based UVES workflow

For the user who is keen on starting reductions without being distracted by detailed documentation, we describe the steps to be performed to reduce the science data provided in the UVES demo data set supplied with the `Reflex 2.11.5` release. By following these steps, the user should have enough information to attempt a reduction of his/her own data without any further reading:

1. Start the `Reflex` application:

   ```
   reflex &
   ```

   The empty `Reflex` canvas as shown in Figure 5.1 will appear.

2. Now open the UVES workflow by clicking on `File -> Open File`, selecting first `uves-6.6.1` and then the file `uves.xml` in the file browser. You will be presented with the workflow canvas shown in Figure 5.4. Note that the workflow will appear as a canvas in a new window.

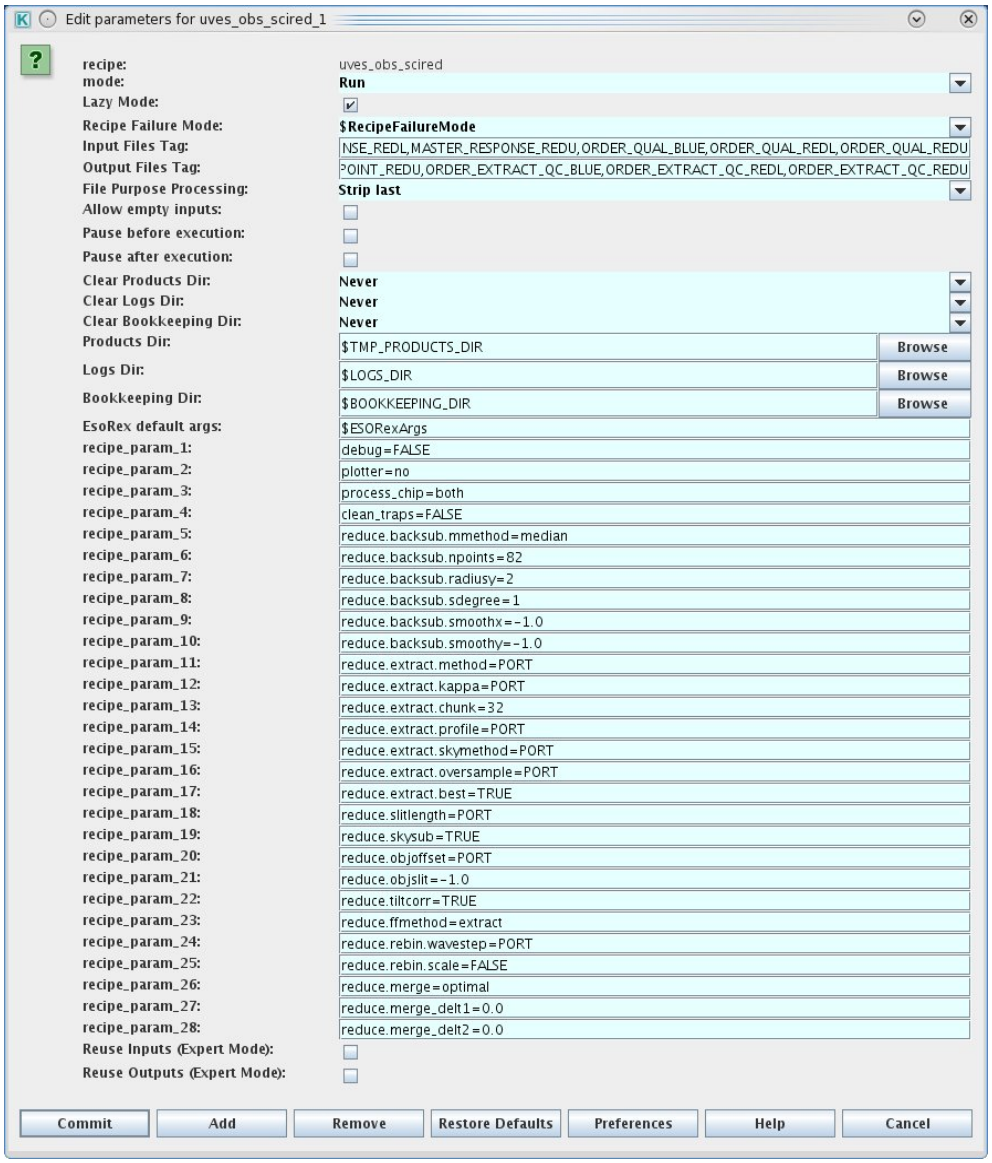


Figure 5.2: *Parameters of a recipe actor. The first group of parameters affect the execution of the pipeline recipe and are common to all recipe actors. The second group of parameters are specific to the pipeline recipe to be called and they are identical to those that can be configured in EsoRex (see 5.2.4).*

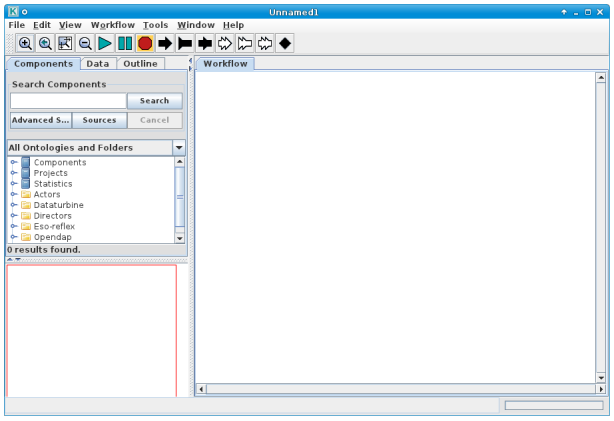


Figure 5.3: *The empty* `Reflex` *canvas.*

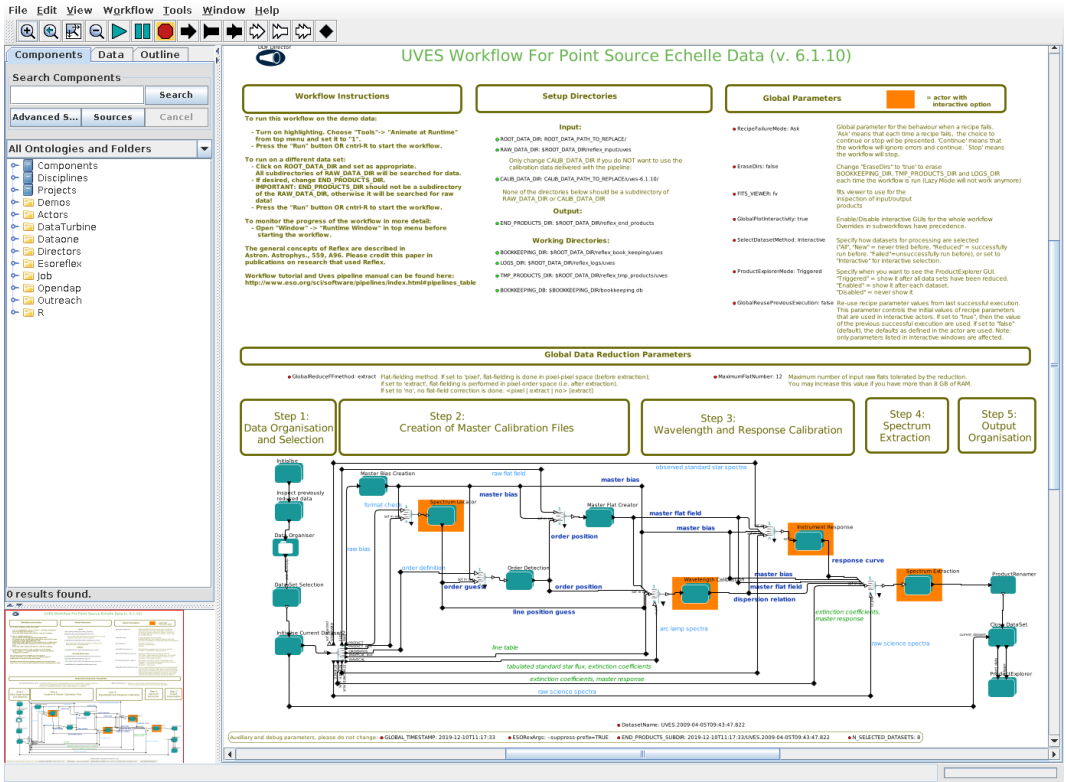


Figure 5.4: *UVES workflow general layout.*

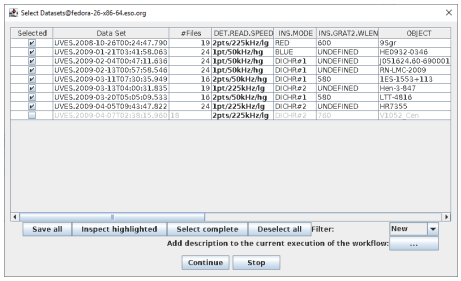


Figure 5.5: *The "Select Datasets" pop-up window. On purpose the demo includes an incomplete (grey) DataSet..*

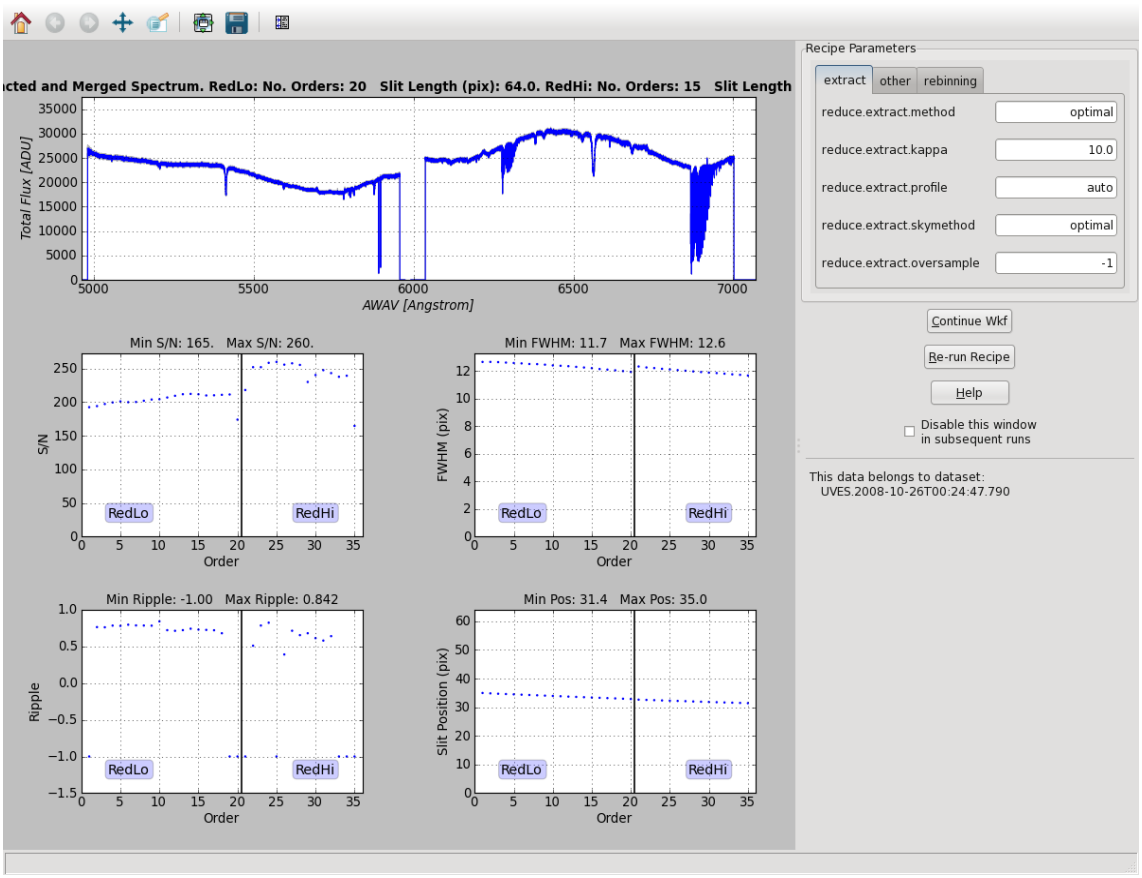


Figure 5.6: *The interactive pop-up window for the* `Spectrum Extraction` *actor and UVES pipeline recipe* `uves_obs_scired`. *The extracted and merged spectrum for the first DataSet is displayed in the top panel, and the flux-calibrated spectrum is displayed in the panel below.*

3. To aid in the visual tracking of the reduction cascade, it is advisable to use component (or actor) highlighting. Click on `Tools -> Animate at Runtime`, enter the number of milliseconds representing the animation interval (100 ms is recommended), and click `OK`.

4. Under "Setup Directories" in the workflow canvas there are seven parameters that specify important directories (green dots). Setting the value of `ROOT_DATA_DIR` is the only necessary modification if you want to process data other than the demo data[2], since the value of this parameter specifies the working directory within which the other directories are organised. Double-click on the parameter `ROOT_DATA_DIR` and a pop-up window will appear allowing you to modify the directory string, which you may either edit directly, or use the `Browse` button to select the directory from a file browser. When you have finished, click `OK` to save your changes.

5. Click the  button to start the workflow

6. The workflow will highlight the `Data Organiser` actor which has recursively scanned the raw data directory (specified by the parameter `RAWDATA_DIR` under "Setup Directories" in the workflow canvas) and constructs the DataSets. Note that the calibration and reference data must be present either in `RAWDATA_DIR` or in `CALIB_DATA_DIR`, otherwise DataSets may be incomplete and cannot be processed. However, if the same reference file was downloaded twice in different places this creates a problem as `Reflex` cannot decide which one to use.

7. The `Data Set Chooser` actor will be highlighted next and will display a "Select Datasets" window (see Figure 5.5) that lists the DataSets along with the values of a selection of useful header keywords[3]. The first column consists of a set of tick boxes which allow the user to select the DataSets to be processed, and by default all complete DataSets are selected.

8. Click the `Continue` button and watch the progress of the workflow by following the red highlighting of the actors. A window will show which DataSet is currently being processed.

9. When the workflow has finished executing the final pipeline recipe `uves_obs_scired` in the `Spectrum Extraction` actor for the first DataSet, an interactive window will appear (see Figure 5.6) which shows a plot of the extracted and merged spectrum in the top panel, and the flux-calibrated version in the panel below. Using the buttons at the top of this window, one may pan and zoom in on the spectrum in order to inspect absorption/emissions lines and other interesting spectral features. Of the other plots in this window, the most important one is the extracted S/N of the spectrum as a function of the spectral order as determined from the CCD noise model (and extraction procedure).

10. When the reduction of the current DataSet finishes, a pop-up window will appear showing the directory were the final products have been saved.

11. The workflow will continue with the remaining DataSets following the same steps described above.

12. After the workflow has finished, all the products from all the DataSets can be found in a directory under `END_PRODUCTS_DIR` with the named with the workflow start timestamp. Further subdirectories will be found with the name of each DataSet.

[2] If you used the install script `install_reflex`, then the value of the parameter `ROOT_DATA_DIR` will already be set correctly to the directory where the demo data was downloaded.

[3] The keywords listed can be changed by right-clicking on the `DataOrganiser` Actor, selecting `Configure Actor`, and then changing the list of keywords in the second line of the pop-up window. Make sure that the `Lazy Mode` is not active and then click on `Commit` to save the change.

5.2.3 Using *Gasgano*

Another convenient tool useful for familiarizing oneself with the UVES pipeline recipes and their usage is the graphical user interface *Gasgano*. It provides a complete graphical user interface for data browsing, classification and association, and offers several other utilities such as easy access to pipeline recipes, documentation and the 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 explicitly passed as a first optional argument the UVES preferences file which defines the proper defaults for UVES data reduction. The user may prefer to set a UNIX alias accordingly. The *Gasgano* main window will appear. In Figure 5.7, a view of a set of UVES data is shown as an example. *Gasgano* can be configured to point 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. Next to each file name, the classification and values of the following FITS keywords are shown (we omit the prefix HIERARCH.ESO):

Keyword name	Purpose
CLASSIFICATION	Data classification
OBS.TARG.NAME	Observation Block target name
EXPTIME	Exposure time
DATE	Observation date
DET.CHIPS	Number 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 header keyword, 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 header 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. 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 left clicking on its name: the corresponding FITS file header will be displayed in 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. By selecting the *Preferences* fields one may set the file filter. This should point to the $HOME/gasgano/config/uves.oca file. This rule file provides simple filtering rules to select data corresponding to a given standard data reduction setting appropriate for the given instrument, in our case UVES.

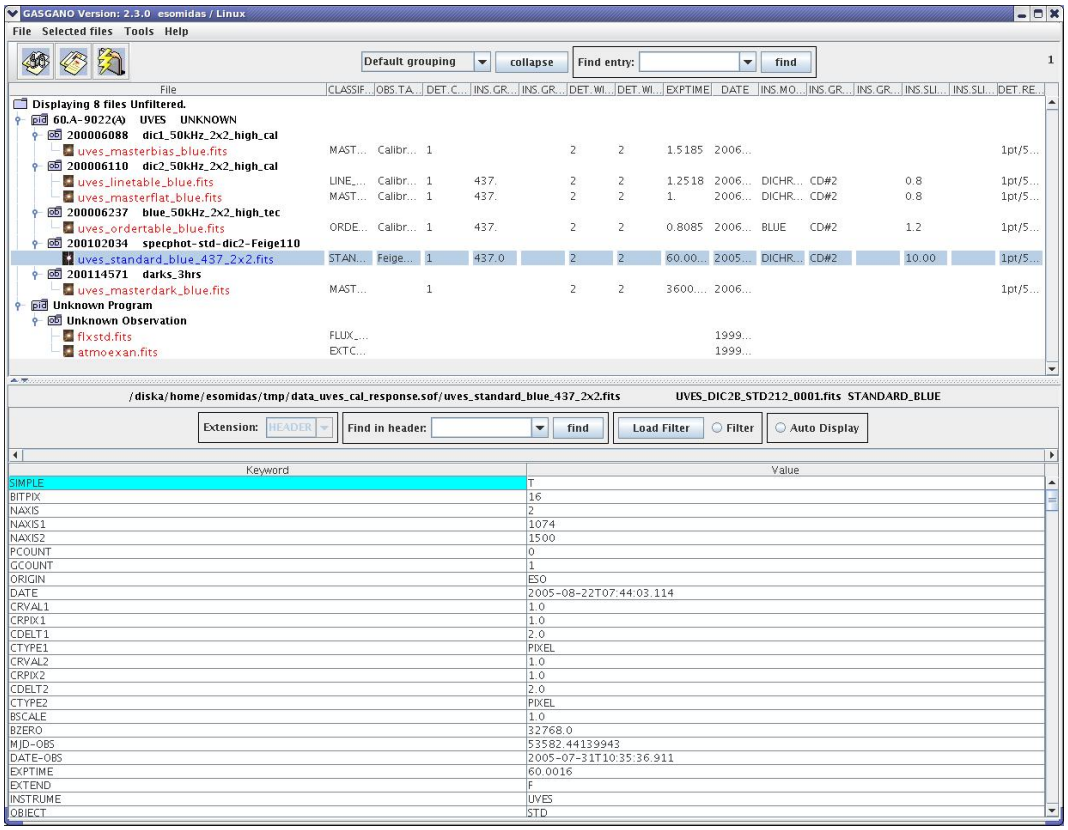


Figure 5.7: *The Gasgano main window.*

Frames can be selected from the main window for processing by the appropriate recipe: in Figure 5.8, the standard star frame, previously produced master bias and master flat frames, together with a line and an order table, a table with the reference standard star spectrum and a table with the atmospheric dispersion, are all selected and sent to the *uves_cal_response* recipe. This action will open a *Gasgano* recipe execution window (see Figure 5.9), 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 modified in the *Parameters* panel (on top). The window contents may be saved for later use by selecting the *Save Current Settings* entry from the *File* menu, as shown in the figure.

At this point the recipe can be launched by clicking the *Execute* button. Messages from the running recipe will appear on the *Log Messages* panel at the bottom, and in case of successful completion, the products will

be listed in the *Output Frames* panel, where they can be easily viewed and located back in the Gasgano main window. To produce useful plots, the user must set the **plotter** recipe parameter value to 'gnuplot -persist' (and have a valid installation of gnuplot package, and the gnuplot command available in the PATH). Please refer to the *Gasgano User's Manual* [12] for a more complete description of the *Gasgano* interface.

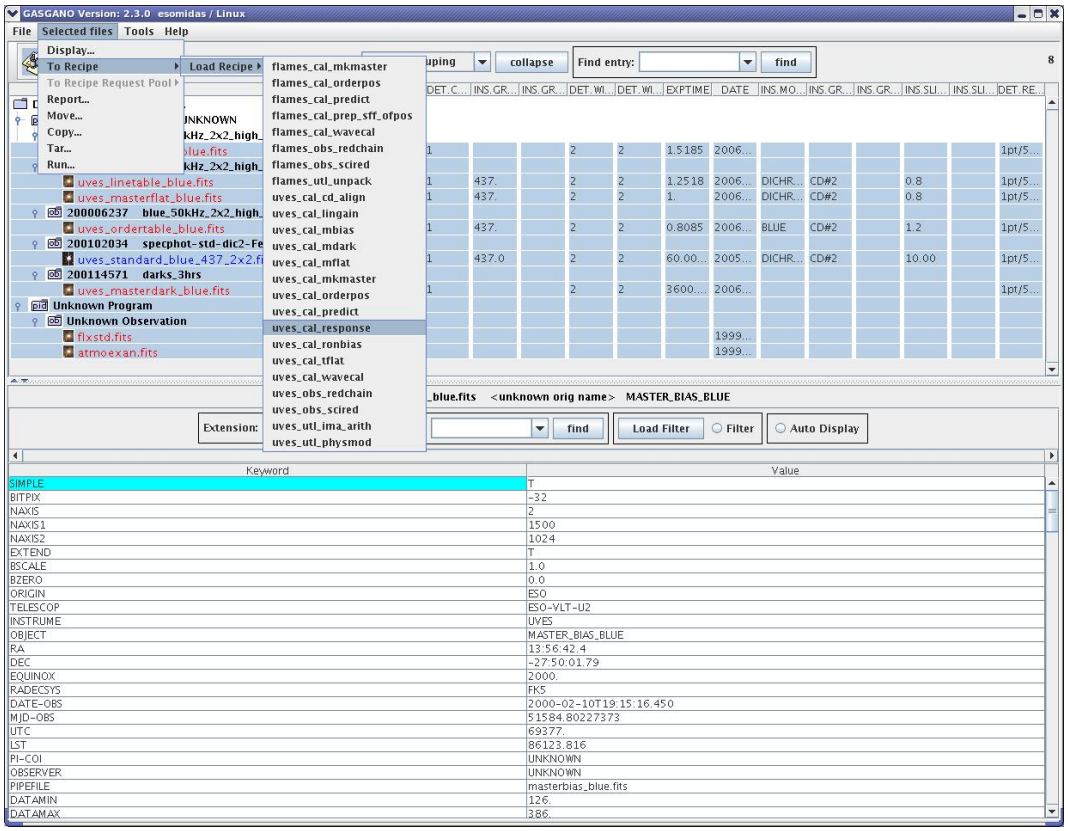


Figure 5.8: *Selecting files to be processed by a UVES pipeline recipe.*

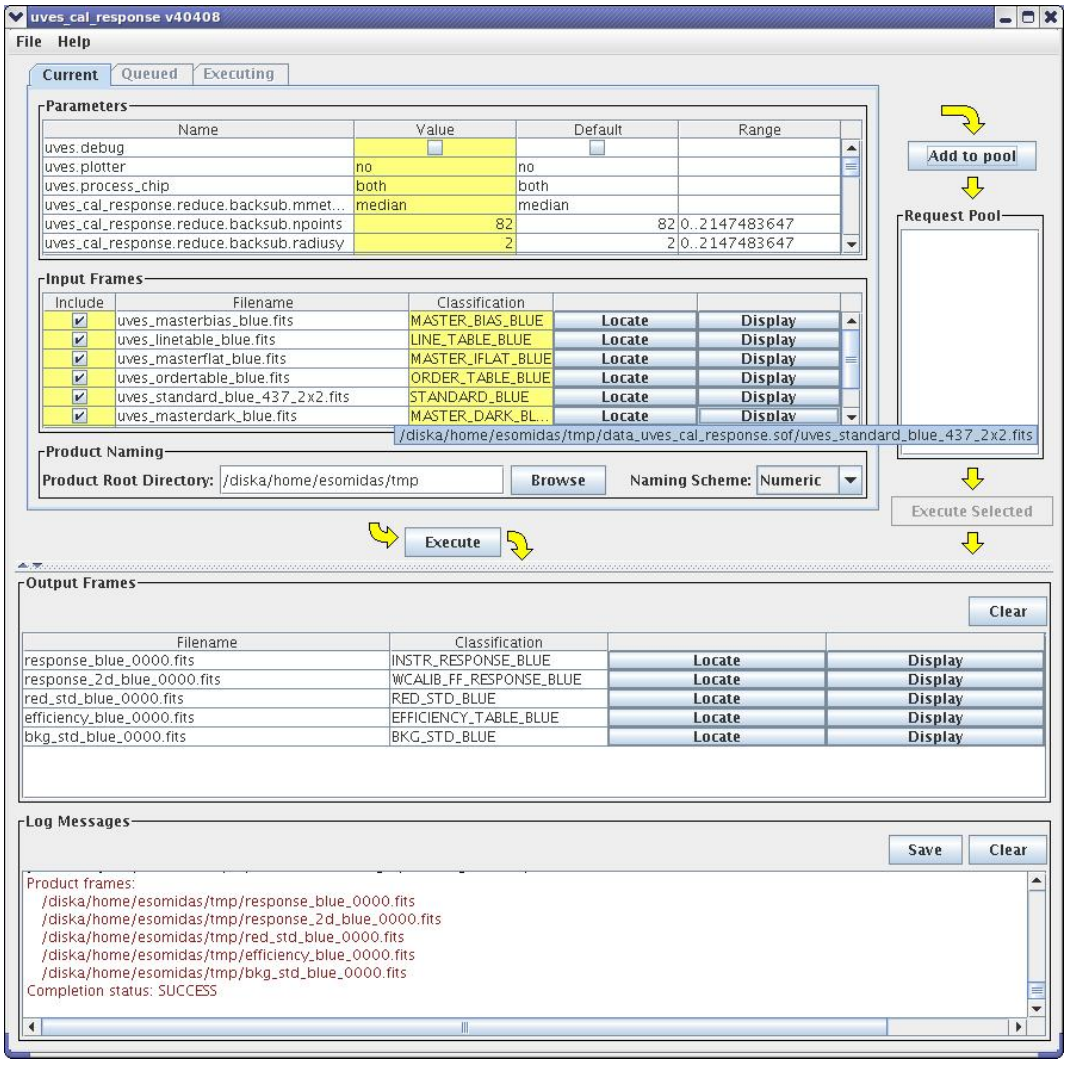


Figure 5.9: *The Gasgano recipe execution window.*

5.2.4 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. A disadvantage of EsoRex is that it does not offer all the facilities available with *Reflex*, and the user must classify and associate the data using the information contained in the FITS header keywords (see Section 7, page 42). The user is also responsible for 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 file names must be listed together with their DO category [4] in an ASCII file called the *set-of-frames* (SOF), which is required when launching a recipe. [5] DO categories for the supported UVES input raw frames are indicated in Section 7.9.

Here is an example of 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/master_bias_blue.fits              MASTER_BIAS_BLUE (optional)
/path_pro/master_flat_blue.fits              MASTER_FLAT_BLUE (recommended)
/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 an 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/master_bias_redl.fits              MASTER_BIAS_REDL (optional)
/path_pro/master_flat_redl.fits              MASTER_FLAT_REDL (recommended)
/path_pro/ordertable_redu.fits               ORDER_TABLE_REDU
/path_pro/lineguess_redu.fits                LINE_GUESS_TAB_REDU
/path_pro/master_bias_redu.fits              MASTER_BIAS_REDU (optional)
/path_pro/master_flat_redu.fits              MASTER_FLAT_REDU (recommended)
/path_ref/thargood_3.fits                    LINE_REFER_TABLE
/path_ref/line_intmon.fits                   LINE_INTMON_TABLE (for operations)
```

[4] 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*

[5] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 5.9, page 29).

The reason for this lack of verification 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 *Reflex* 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 5.2.2, page 21). Also, this lack of verification 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 for using EsoRex is as follows:

esorex [esorex_options] recipe_name [recipe_options] set_of_frames

To obtain more information on how to customise EsoRex (see also [11]) run the command:

esorex --help

To generate a configuration file esorex.rc in the directory $HOME/.esorex, execute 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 obtain a brief description of each parameter for a specific pipeline recipe, execute the command:

esorex --help recipe_name

To obtain more details about a given recipe, issue the following command:

esorex --man-page recipe_name

Recipe configuration: Each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe.[6] 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`.

The command

esorex --create-config recipe_name

generates a default configuration file **recipe_name.rc** in the directory **$HOME/.esorex**[7].

For example, the recipe *uves_cal_wavecal* has its *EsoRex* generated configuration file named `uves_cal_wavecal.rc`, and is generated with the command:

[6] The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 5.9, page 29).

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

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
```

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.

A recipe configuration file different from the default one may 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 specifying its name to *EsoRex*, together with the name of a set-of-frames. For instance, the following command would be used to run the recipe *uves_cal_wavecal* to process the files specified in the set-of-frames `uves_cal_wavecal.sof`:

esorex uves_cal_wavecal uves_cal_wavecal.sof

The recipe parameters may be modified either by directly editing the configuration file that is used, 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 command should be entered:

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 a more detailed inspection of the recipe execution.

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 [11].

5.2.5 Example of data reduction using EsoRex

Here we provide an example of data reduction for data obtained with the UVES BLUE arm using EsoRex. The data reduction of data obtained with the red arm is similar. The only difference in the SOF file is that the data tag: BLUE has to be replaced by RED. Moreover, twice as much calibration data, corresponding to the lower and the upper chips, and with similar tags but suffix REDL and REDU, are required.

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/uves_paranal_extinct_model.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 the 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), it 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. To do this for the biases for instance, replace the five entries in the SOF for the raw bias frames by the single entry for the master bias frame.

In the following a typical step-by-step data reduction procedure is described. Figure 5.10 gives an overview.

We suggest that the user groups the data according to detector arm, dichroic and detector binning setting. In the examples below, we assume that the user is in possession of data acquired with the BLUE arm. `/path_ref/`

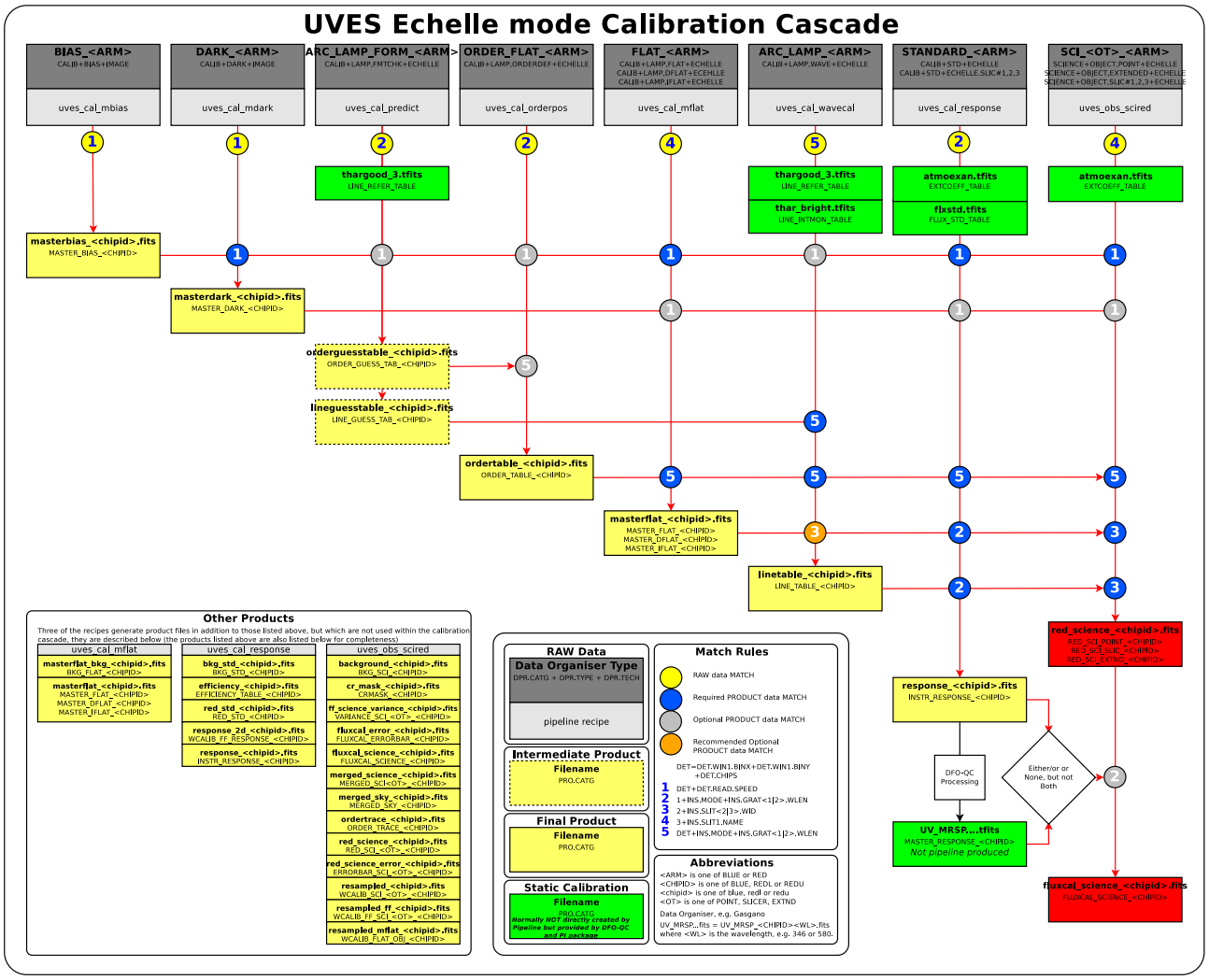


Figure 5.10: *The UVES calibration cascade.*

indicates the full path to the directory containing the reference ancillary data, and `/path_pro/` indicates the full path to the source tree directory containing the product data.

5.3 FITS header keywords for raw data identification

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

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

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

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

Arc lamp frames for computing 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
```

In order to obtain additional information on the overall telescope and instrument response, the user may also want to reduce 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 the pipeline product filenames are the original as set by the pipeline. This corresponds to having the parameter **esorex.caller.suppress-prefix** in the EsoRex configuration file (`$HOME/.esorex/esorex.rc`) set to TRUE. Otherwise EsoRex will rename the pipeline products using a common prefix (set by the parameter **esorex.caller.output-prefix**), a four digit increasing number, and terminating the FITS file with the extension ".fits". We suggest verifying that in the EsoRex configuration file the flag *readonly* is set to FALSE, if the user would like to run the same recipe several times with EsoRex having predefined values for product files. This setting allows the pipeline to overwrite previously generated products [8].

1. Select the raw biases and list them in an ASCII file uves_cal_mbias.sof:

[8] 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 extension .fits.

```
/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	image	MASTER_BIAS_BLUE	master bias

mv *.fits /path_pro

2. Now select 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	image	MASTER_DARK_BLUE	master dark

mv *.fits /path_pro

3. Generate guess order and line tables. Format check frames are listed together with the required calibration frames in an ASCII file, uves_cal_predict.sof. This file will look like following:

```
/path_raw/uves_arc_lamp_form_blue.fits      ARC_LAMP_FORM_BLUE
/path_ref/thargood_3.fits                   LINE_REFER_TABLE
/path_cdb/masterbias_blue.fits              MASTER_BIAS_BLUE   (optional)
```

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 three files (in the following table FITS files have 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

mv *.fits /path_pro

4. 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
/path_pro/orderguesstable_blue.fits ORDER_GUESS_BLUE    (recommended)
/path_cdb/masterbias_blue.fits      MASTER_BIAS_BLUE    (optional)
```

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

mv *.fits /path_pro

5. Then one selects the raw flat fields and list them in an ASCII file uves_cal_mflat.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/masterdark_blue.fits MASTER_DARK_BLUE  (optional)
/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	image	MASTER_FLAT_BLUE	master flat
masterflat_bkg_blue	image	BKG_FLAT_BLUE	inter-order background

mv *.fits /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 (recommended)
```

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

The command

esorex uves_cal_wavecal uves_cal_wavecal.sof

will generate the following product:

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'
```

mv *.fits /path_pro

7. As an optional step, a standard star may be reduced in order to obtain the efficiency of the system telescope+instrument+detector. The standard star raw frame name should be stored in the following ASCII file uves_cal_response.sof together with the 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/uves_paranal_extinct_model.fits EXTCOEFF_TABLE
```

The command:

esorex uves_cal_response uves_cal_response.sof

will generate the following products among others:

default recipe filename	format	PRO.CATG	short description

response_blue	spectra 1D	INSTR_RESPONSE_FINE_BLUE	instrument response
red_std_blue	spectra 1D	RED_STD_BLUE	extracted, merged std star spectrum
efficiency_blue	table	EFFICIENCY_TABLE_BLUE	instrument efficiency
bkg_std_blue	image	BKG_STD_BLUE	sky background

mv *.fits /path_pro

8. Finally, the raw science frame is reduced. The raw science frame is listed together with the 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/response_blue.fits              INSTR_RESPONSE_FINE_BLUE (optional)
/path_ref/uves_paranal_extinct_model.fits EXTCOEFF_TABLE (optional)
```

The frames tagged as INSTR_RESPONSE_FINE_BLUE and EXTCOEFF_TABLE are optional, and are used to flux calibrate the reduced science spectra. If the user wishes to perform flux calibration, then both of these files must be supplied.

The command:

esorex uves_obs_scired uves_obs_scired.sof

will generate the following products among others:

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 spectrum
merged_science_blue	1d (wav) image	MERGED_SCIENCE_BLUE	extracted, flatfielded, wavelength calibrated, merged science spectrum
error_red_science_blue	1d (wav) image	ERRORBAR_SCIENCE_BLUE	uncertainty on RED_SCIENCE_BLUE
fluxcal_science_blue	1d (wav) image	FLUXCAL_SCIENCE_BLUE	flux calibrated version of RED_SCIENCE_BLUE
fluxcal_error_science_blue	1d (wav) image	FLUXCAL_ERRORBAR_SCIENCE_BLUE	uncertainty on FLUXCAL_SCIENCE_BLUE

6 Known problems

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

- This pipeline release has been updated to support data reduction for the new RED ARM CCD (but still allow reduction of older data). The number of possible UVES instrument settings is, in principle, infinite (the wavelength setting can change continuously). Hence we have limited the pipeline verifications to a finite number of settings as indicated in Section 3. The new RED ARM detector has a slightly different position and orientation with respect to the previous detector. In order to reduce the new data, the pipeline must set appropriate values for the offset parameters: **ccd_rot_angle_off**, **trans_x** and **trans_y**. We have been able to verify that the change in the parameter **ccd_rot_angle_off** is uniform with wavelength. Because the instrument is non-linear, and the detector configurations at the moment of the upgrade were slightly shifted in the cross order direction with respect to the reference position from the year 2000, the pipeline requires **trans_x** and **trans_y** parameter values that depend on the wavelength setting. For this reason, in the case of non-standard wavelength settings, the user may need to find the optimal value of the x and y translation parameters as follows.

  After execution of the recipe uves_cal_predict, the offsets QC.MODEL.DIFFXMED, QC.MODEL.DIFFYMED may be greater than 4-6 pixels, in which case the user may adjust the values of the parameters **trans_x** and **trans_y** to minimize the absolute values of QC.MODEL.DIFFXMED, QC.MODEL.DIFFYMED.

  In most of the cases it is sufficient to set **trans_x** to QC.MODEL.DIFFXMED, and **trans_y** to QC.MODEL.DIFFYMED and verify that in the repeat run of uves_cal_predict the corresponding QC parameters decrease in absolute value. This procedure may be iterated so that they ideally reach the value of 0.

  By running uves_cal_predict with the option **debug** set to TRUE, the recipe generates additional products. The user may want to verify that the reference line catalog predicted positions (XMOD, YMOD) provide good matches with the actual line positions on the format-check frame. For this comparison the user should use the products pline_redl.fits and arclampform_redl.fits (same for the "redu" corresponding FITS file). If the user uses gaia or rtd (and Data-Servers–>Local Catalogs–>Load from file...) it is necessary to rename the columns XMOD, YMOD to x,y to have proper display.

- 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 on the scale of one order and on the scale of the inteference patterns.

- The recipe uves_cal_mflat due to hardware RAM limts, may combine only a given number of frames. On an Intel core i5 with 3GB RAM the measured maximum are: 25 BLUE arm 1x1 frames, or 11 RED arm 1x1 frames. This number increases linearly with the bin size in pixels.

- Occasionally the recipe uves_cal_wavecal may fail with error like:

```
WaveCal: Dumping all 7 error(s):
WaveCal:   [1/7] 'Singular matrix: 25 coefficients. Only 24 points'
                 (17) at
```

```
                        uves_polynomial_fit_2d:uves_utils_polynomial.c:1557
WaveCal:   [2/7] 'Singular matrix: Could not fit polynomial' (17) at
                           uves_polynomial_regression_2d:uves_utils.c:3085
WaveCal:   [3/7] 'Singular matrix: Error fitting polynomial. Possible
                  cause: too few (81) line identifications' (17) at
                  calibrate_global:uves_wavecal_identify.c:1136
WaveCal:   [4/7] 'Singular matrix: Could not perform global
                  calibration' (17) at
                  uves_wavecal_identify:uves_wavecal_identify.c:327
WaveCal:   [5/7] 'Singular matrix: Could not calibrate orders' (17)
                  at uves_wavecal_process_window:uves_wavecal_body.c:720
WaveCal:   [6/7] 'Singular matrix: Wavelength calibration failed'
                  (17) at uves_wavecal_exe_body:uves_wavecal_body.c:1913
WaveCal:   [7/7] 'Singular matrix: WaveCal execution failed' (17) at
                  uves_cal_wavecal:uves_wavecal_impl.c:214
esorex: Execution of recipe 'uves_cal_wavecal' failed, status = 17
```

This may be due to the fact that the frame may have line FWHM very different from the usual for example because the frame have been acquired with a slit width much larger than the default. In this case we suggest the user either to use a different data set with standard slit width values or to increase (significantly, eventually up to 10 pix) the parameter **tolerance**. In this last case we warn the user that the corresponding wavelength solution may be not very accurate (being accepted outliers up to **tolerance** pixels from the solution.

- Line tilt correction (correcting for the fact that the dispersion relation depends on slit position) is not supported for 2d spectrum extractions, but only for average/linear/optimal extractions.
- The recipe uves_obs_scired reduces only the first science frame listed in the corresponding SOF. In order to reduce several science frames, the recipe should be run several times.
- If, in the reduced spectra generated by the recipe uves_obs_scired, there are still residual cosmic rays present, than we suggest that the user decreases the value of **uves.reduce.kappa**.
- Sometimes, setting the uves_cal_wavecal recipe parameter **degree** to 5, the user may find artifacts (absorption lines appearing as doublets) in the merged spectrum. For this reason we set the parameter default value to 4. The user may also leave the pipeline to find the best polynomial degree to have the best wavelength calibration accuracy. This is possible by setting a negative value for the **degree** parameter.
- 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 in the final spectrum error bars. For this reason the default value **reduce.ffmethod=extract** is recommended.
- Artifacts present in merged spectra of RED760 (or beyond) settings. Can this be removed?

  For data observed in the red settings (760 or beyond), we recommend to use `reduce.ffmethod = "pixel"` (together with `reduce.merge_delt1=14` and `reduce.merge_delt2=4`). Thereby the user can avoid artefacts from residual flat-field fringes that have a noise-like structure and might dominate the spectral structure and compromise the signal-to-noise ratio. The drawback of that method are short spectral gaps between the orders in the REDU range.
- In case the recipe uves_cal_response does not find the reference standard star in the input FLUX_STD_TABLE catalog, this may be due to increased proper motion of the standard star, in which case we recommend to increase the value of the parameter **paccuracy** for example from 60 to 70 [arcsec].

7 Instrument Data Description

7.1 Format check frames

A format check frame is an image taken using a ThAr lamp and a narrow slit. This frame is used in combination with a physical model of UVES, the information contained in the format check frame FITS header and in a ThAr reference line table, to find a "guess" solution for the spectral format (order locations and wavelength calibration). This allows the user later on to obtain robust and automatic spectral format solutions. The format check frame is a recommended, but optional, calibration file.

7.2 Bias frames

Bias frames give the read out of the CCD detector for zero integration time with the shutter closed. Usually they are taken as a set of five exposures from which, through stacking, a master bias frame is created. The master bias frame contains the bias offset for the detector, including any spatial dependence.

7.3 Dark frames

For the UVES detectors, dark frames are obtained occasionally, as long exposures with the shutter closed. They are used to measure the CCD dark current. Dark frames are taken for 1x1 and 2x2 binnings with typical exposure times of 1h. Dark frames using an open shutter are also taken (since December 2001). The open-shutter dark frames include in addition to the CCD dark current, the flux contribution from the camera enclosure.

Typical dark current values are reported on the ESO Website under
www.eso.org/observing/dfo/quality/UVES/qc/dark_qc1.html. The UVES detector dark currents may be considered negligible to a first approximation, and they can usually be excluded from the data reduction chain.

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 precisely marking the order locations.

7.5 Flat field frames

Flat field frames are long slit exposures taken with a continuum lamp. They provide information on the response of the detector, mapping the efficiency variations of the detector at all scales from pixel-to-pixel variations, through to large scale blazing function variations, and including intermediate scale variations caused by fringing (in the far red). Usually flat frames are taken as a set of five frames which, after bias (and possibly dark) frame subtraction, are combined into a master flat frame with rejection of outliers such as cosmic ray events. The final master flat field is also background subtracted in order to eliminate diffuse 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 internal flat-field lamp flats. The principal difference between the TFLATs and normal FLATs is that the power supply to 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. The internal lamp flats have the advantages that they can be acquired during day time, they isolate the instrument performance from that of the telescope and the transmission of the atmosphere, and they can be taken to for various instrument configurations thus allowing one to monitor the performance of the individual components of the instrument.

TFLATs are NOT suitable or intended for scientific reductions.

7.7 Arc frames for wavelength calibration

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

7.8 Standard star frames

Standard star frames are observations of standard stars for which the source spectra is known. Such observations enable the determination of the following:

- The response curve (i.e. the conversion between a science spectrum and a flux calibrated spectrum). The response curve provides a better relative flux calibration than that provided by the flat field lamp, because the flat field correction includes the spectral energy distribution of the flat field lamp.
- The overall efficiency (DQE) of the telescope+instrument+detector system (corrected for atmospheric extinction). This function is usually evaluated for the purpose of trending.

An observer, by employing the calibration frames described above, may reduce the following types of science data:

- Point-like sources. The UVES pipeline has been designed to extract the spectra of this kind of source.
- Extended sources. These sources may be extracted by setting the corresponding parameter in the extraction method (see 10.10.4)
- Multi-object sources. For spectra of multiple objects observed through the same slit, the UVES pipeline may extract the individual spectra so long as the multiple objects are not blended in any way.

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 these data may be found at *https://www.eso.org/observing/dfo/quality/UVES/reports/RAWDISP/*

REFERENCE/rawfiles_START_ALL.html. The different frame types can be identified by the values of the DPR header keywords in their FITS headers (see [18]). These keywords are generated by the UVES templates (for a description of the UVES templates see [6]). A given frame type may be processed by a single or multiple pipeline recipes. The individual pipeline recipes are described in Section 11.2. In most cases, reference data frames are needed to reduce a given frame. These reference data have to match the input frame for a number of instrument parameters (e.g. to apply a flat field correction to a science frame, only a flat field frame taken with the same central wavelength, same slit length, etc. will be suitable for the correction). In the following, 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_x_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 7.1 (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 7.1 (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 7.1 (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 7.1 (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 7.1 (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 7.1 (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`

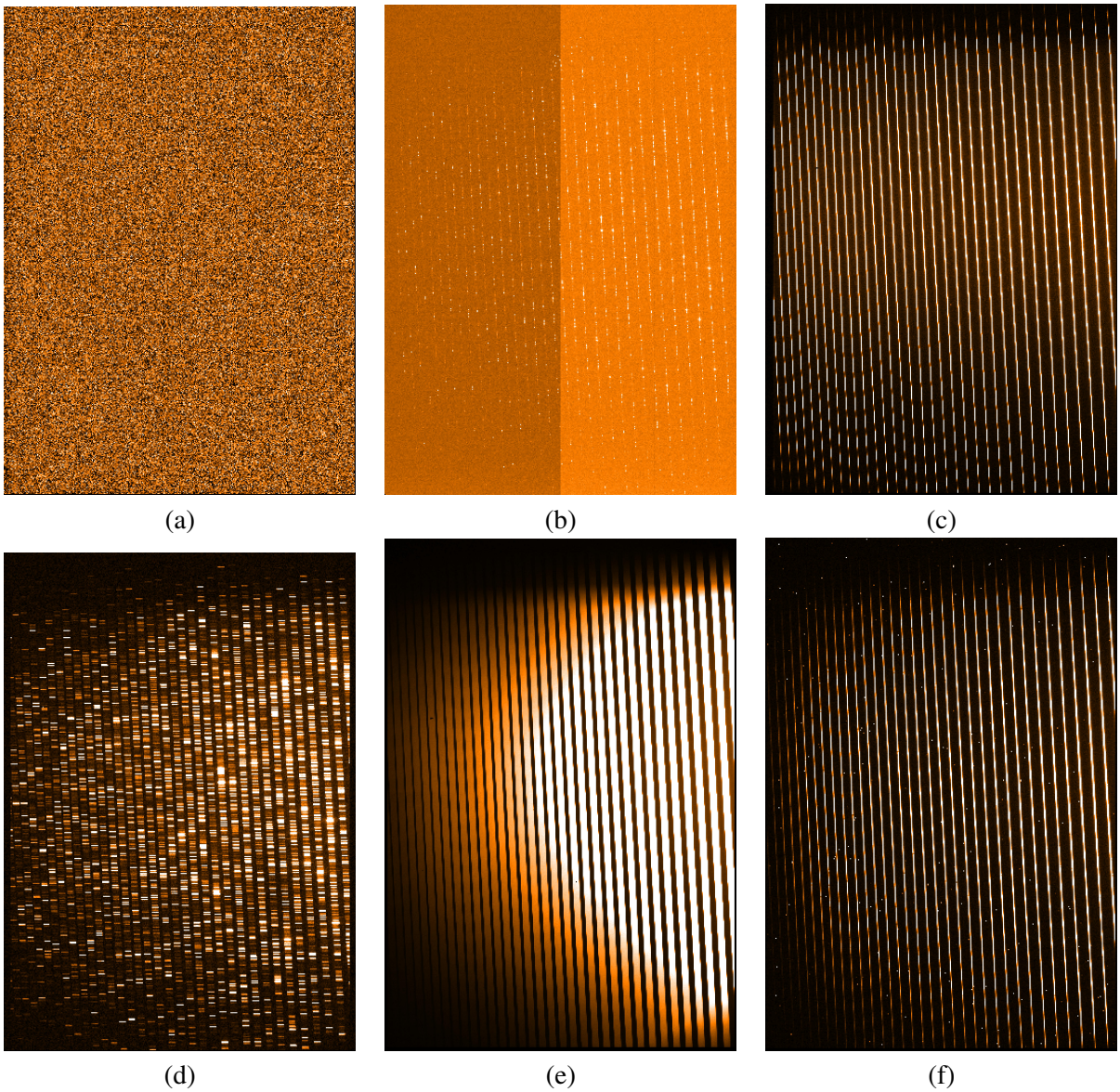


Figure 7.1: (a) a raw bias frame; (b) a raw format check frame; (c) a raw order frame; (d) a raw arc lamp frame; (e) a raw flatfield frame; (f) a raw std star frame.

8 Static Calibration Data

In this section, the ancillary data required for UVES data reduction are described. For each of the data tables, we indicate the corresponding value of the HIERARCH ESO PRO CATG (in short PRO.CATG) FITS header keyword that must be used to identify the relevant file in the *Set of Frames* (see Section 5.2.4, page 30). More information on these ancillary data may be found at *https://www.eso.org/sci/observing/phase3/data_types.html*

8.1 Emission line reference table

A reference list of arc lines is necessary to perform the wavelength calibration. The corresponsing value of PRO.CATG is LINE_REFER_TABLE. The wavelength values in the reference table refer to air. This table is an input to the recipes uves_cal_predict and uves_cal_wavecal.

In the package distributed with the pipeline we provide two: thargood_2.fits and thargood_3.fits. The first for historical reasons, it was the first used during UVES commissioning and operations, the second, revised in early 2007, is more accurate and recommended.

8.2 Standard stars flux table

This table contains photometrically aligned model spectra for a given list of standard stars in units of $erg \cdot cm^{-2} \cdot s^{-1} \cdot$ Å^{-1}. The PRO.CATG is FLUX_STD_TABLE and it contains data for seven flux standards: GD71, Feige 110, GD153, LTT3218, LTT7987, EG21, and EG274[9].

8.3 Atmospheric extinction table

This table provides the extinction coefficient as a function of the wavelength expressed in Å. The curve compiled by Patat et al. (A&A, 527, A91, 2011) has been replaced by a typical LBLRTM model spectrum for Paranal, which covers the range 3000 Å–10994 Å. We interpolated across the regions of strong telluric absorption in the VIS range (5855 Å–5992 Å, 6261 Å–6349 Å, 6438 Å–6600 Å, 6821 Å–7094 Å, 7127 Å–7434 Å, 7562 Å–7731 Å, 7801 Å–8613 Å, 8798 Å–10338 Å, $>$10500 Å). Since the model is not an actual fit to the data we added 0.03 mag (well below the measurement uncertainties) to the extinction values to get them better aligned to the FORS2 data points reported by Patat et al. (2011). 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 uniformly distributed on the blue/red detectors can be provided by DFO and has PRO.CATG equal to LINE_INTMON_TABLE.

[9]The reference data for the old UVES standard stars are available at *https://ftp.eso.org/pub/dfs/pipelines/instruments/uves/uves_response_old_calib.tar.gz*. Please be aware that these data are not well suited for the response determination due to their coarse sampling and low resolution.

8.5 Master response curves

For the standard set-ups master response were created which allow the user to flux-calibrate their data even if no suitable standard star observation exists. These data have the `PRO.CATG` value `MASTER_RESPONSE_<ARM>` (see Section 1.4 for a description of the `_<ARM>` suffix) and are delivered by calSelector when downloading data from the archive.

8.6 Table with wavelength points to fit the response

The UVES pipeline uses stellar model spectra as reference spectra to determine the response from flux standard star observations. In some cases the line cores in these reference spectra deviate on a 1-2% level from the actually observed ones. Thus the ratio of reference to observed spectrum may show residuals at the places of strong lines. In order to avoid such residuals and regions of strong telluric absorption the pipeline uses pre-defined points along the spectrum to fit the response (PRO.CATG = RESP_FIT_POINTS_CATALOG)[10].

[10]For the old UVES standard stars a table with preliminary wavelengths (optimized for high-resolution reference data of hot white dwarf stars) can be found at *https://ftp.eso.org/pub/dfs/pipelines/instruments/uves/uves_response_old_calib.tar.gz*. Please be aware that these fit points need to be adapted for the specific standard star and the suitability of the response will depend on the quality of the reference spectrum.

9 Data Reduction

In this section, we offer an overview of the main problems the data reduction needs to solve, and then we list the required data and recipes required to solve them, providing 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 orders need to be traced in a robust manner.
- The detector pixel-to-pixel sensitivity variations, and the blaze function, need to be measured and corrected for.
- The wavelength calibration of the spectral data must be performed in a robust and automatic manner.
- The telescope+instrument+detector system efficiency should be determined.
- The science data should be corrected and calibrated for all the above effects.

9.2 Required input data

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

- Raw frames:
  - Format check frames to determine guess order and line tables.
  - Order definition frames to determine the order tables.
  - Arc lamp frames to determine the line table and wavelength solution.
  - 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 which provides stable order tracing.

  - Guess line table which will help to define an automatic, accurate and stable wavelength calibration.
  - Master bias and master dark frames to subtract from the flat frames, standard star frames and science frames.
  - Master flat field frame to correct for different detector pixel efficiencies, the blaze function, and the detector fringing at longer wavelengths.
  - Line tables to calibrate the object and sky spectra in wavelength.
  - Instrument response to calibrate merged spectral data in flux units.
- Reference files:
  - Reference line table for use in producing the guess and final line tables.
  - Atmospheric extinction table to be used in the instrument efficiency computation.
  - Standard star fluxes to be used in the instrument efficiency computation.
  - Table with wavelength points to fit the response.

9.3 Reduction cascade

The UVES data reduction follows the sequence shown in Figure 5.10.

In the text below, the DO category corresponding to a frame is written in parentheses and often ends with a _<...> suffix (whose meaning is described in Section 1.4).

- Create a master bias frame (MASTER_BIAS_<CHIPID>) with the recipe **uves_cal_mbias** from a set of raw bias frames (BIAS_<ARM>) .
- Create a master dark frame (MASTER_DARK_<CHIPID>) with the recipe **uves_cal_mdark** from a set of raw dark frames (DARK_<ARM>) and a master bias frame.
- Generate a guess order (ORDER_GUESS_TAB_<CHIPID>) and guess line table (LINE_GUESS_TAB_<CHIPID>) with the recipe **uves_cal_predict** taking as input a format check frame (ARC_LAMP_FORM_<ARM>) and a reference catalogue of arc lines (LINE_REFER_TABLE). The MASTER_BIAS_<CHIPID> frame is an additional optional input.
- Refine the order guess table into an order table (ORDER_TABLE_<CHIPID>) with the recipe **uves_cal_orderpos** taking as input an order definition frame (ORDER_FLAT_<ARM>) obtained by illuminating the UVES narrow slit with a continuum lamp. The ORDER_GUESS_TABLE is an additional optional input, recommended to make sure that the same detector area covered by the orders predicted by the model is also analyzed in the detailed order tracing performed by the uves_cal_orderpos recipe.
- Create a master flat frame (MASTER_FLAT_<CHIPID>) with the recipe **uves_cal_mflat** from a set of raw flat frames (FLAT_<ARM>), a master bias frame and a master dark frame (optional). Additional input is the order table (ORDER_TABLE_<CHIPID>).
- Refine the guess line table into a line table (LINE_TABLE_<CHIPID>) containing the wavelength solution for three traces, the object trace and two sky traces, using the recipe **uves_cal_wavecal**. The required inputs are an arc lamp frame

(ARC_LAMP_<ARM>) obtained illuminating the UVES wide slit with an arc lamp, the order table, the guess line table, and the reference line catalogue. We recommend also to use the master bias and the master flat, although these inputs are optional.

- Determine the instrument response (INSTR_RESPONSE_FINE_<CHIPID>) and telescope + instrument + detector efficiency (EFFICIENCY_<CHIPID>) with the recipe **uves_cal_response** using as input 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 the science data (SCIENCE_<ARM>) with the recipe **uves_obs_scired** using a master bias, a master dark, a master flat, an order table, and 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 in automatic mode, one may use the **uves_obs_redchain** recipe on a complete set of raw and ancillary data.

10 Pipeline Recipe Interfaces

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

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

In the following we assume that the input files already exist as FITS 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`.

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 keyword is PRO.CATG, which is used to classify each frame, and to associate the proper calibration frame to each raw frame.

For a description of the various _<...> suffixes used in the file and PRO.CATG/TAG names used below, refer to Section 1.4.

The raw 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 also able to process pipeline products generated by the MIDAS based pipeline. The reverse 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. We recommend to use recipe parameter defaults unless the specific data reduction case requires particular parameter 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 the usage of different parameter values to reduce data from the two UVES red arm detector chips.

A full description of each parameter is obtained 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.

We also list quality control parameters, which are stored in the corresponding pipeline products. More information on instrument quality control can be found at www.eso.org/qc.

10.1 uves_cal_mbias

The recipe uves_cal_mbias creates a master bias frame.

10.1.1 Input

frame tag/category	nr	filename example
BIAS_<ARM>	+	/path_raw/uves_bias_<arm>.fits

10.1.2 Output

default recipe file name	format	PRO.CATG	short description
masterbias_<chipid>.fits	image	MASTER_BIAS_<CHIPID>	master bias frame

10.1.3 Quality control

The recipe computes the following QC parameters:

QC.DUTYCYCL	Time to store a frame [days]
QC.DUTY.CYCL	Time to complete an exposure, minus actual exposure time
QC.OUTi.RON.RAW	Read noise on raw images [ADU]
QC.OUTi.RON.MASTER	Read noise on master bias frame [ADU]
QC.OUTi.STRUCTY	Structure noise in Y [ADU]

QC.OUTi.STRUCTX	Structure noise in X [ADU]
QC.MEAN.RATIO.REF	Ratio to reference frame
QC.STD.MEAN.RATIO.REF	Sigma of MEAN.RATIO.REF

Additional QC parameters are logged in the power spectrum product:

QC.PD.STD	Raw file: std
QC.PD.STDMAD	Raw file: stdmad
QC.RATIO.PDi	Raw filei: ratio std/stdmad
QC.RATIO.PD.AVG	Average ratio std/stdmad
QC.RATIO.PD.MAX	Maximum ratio std/stdmad
QC.INDEX.PD.MAX	Index of the maximum ratio

10.1.4 Parameters

parameter	alias	default	min	max
uves_cal_mbias.clean_traps	clean_traps	FALSE		
uves_cal_mbias.stack_method	stack_method	median		
uves_cal_mbias.klow	klow	5.	0.	100.
uves_cal_mbias.khigh	khigh	5.	0.	100.
uves_cal_mbias.niter	niter	5	0	100

Note that the default parameters are robust.

10.2 uves_cal_mdark

The recipe uves_cal_mdark creates a master dark frame.

10.2.1 Input

frame tag/category	nr	filename example
DARK_<ARM>	+	/path_raw/uves_dark_<arm>.fits
MASTER_BIAS_<CHIPID>	?	/path_raw/masterbias_<chipid>.fits

10.2.2 Output

default recipe file name	format	PRO.CATG	short description
masterdark_<chipid>.fits	image	MASTER_DARK_<CHIPID>	master dark frame

10.2.3 Quality control

The recipe computes the following QC parameters:

QC.DATANCOM	Number of coadded frames
PRO.DATAMED	Median frame level [ADU]
QC.REGij.MIN	Min of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.MAX	Max of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.AVG	Mean of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.MED	Median of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.RMS	Rms of region i j of size **box_sx** × **box_sy** [ADU]
QC.REG.MIN.MIN	Min of all region Mins [ADU]
QC.REG.MIN.MAX	Max of all region Mins [ADU]
QC.REG.MIN.AVG	Mean of all region Mins [ADU]
QC.REG.MIN.MED	Median of all region Mins [ADU]
QC.REG.MIN.RMS	Rms of all region Mins [ADU]
QC.REG.MAX.MIN	Min of all region Maxs [ADU]
QC.REG.MAX.MAX	Max of all region Maxs [ADU]
QC.REG.MAX.AVG	Mean of all region Maxs [ADU]
QC.REG.MAX.MED	Median of all region Maxs [ADU]
QC.REG.MAX.RMS	Rms of all region Maxs [ADU]
QC.REG.AVG.MIN	Min of all region Means [ADU]
QC.REG.AVG.MAX	Max of all region Means [ADU]
QC.REG.AVG.AVG	Mean of all region Means [ADU]
QC.REG.AVG.MED	Median of all region Means [ADU]
QC.REG.AVG.RMS	Rms of all region Means [ADU]
QC.REG.MED.MIN	Min of all region Medians [ADU]
QC.REG.MED.MAX	Max of all region Medians [ADU]
QC.REG.MED.AVG	Mean of all region Medians [ADU]
QC.REG.MED.MED	Median of all region Medians [ADU]
QC.REG.MED.RMS	Rms of all region Medians [ADU]
QC.REG.RMS.MIN	Min of all region Rms [ADU]
QC.REG.RMS.MAX	Max of all region Rms [ADU]
QC.REG.RMS.AVG	Mean of all region Rms [ADU]
QC.REG.RMS.MED	Median of all region Rms [ADU]
QC.REG.RMS.RMS	Rms of all region Rms [ADU]
QC.DARK.MEAN.RAW	Mean value of raw frame
QC.DARK.MEAN	Mean value across master dark
QC.DARK.CURRENT	DARK.MEAN, normalised to 3600 secs exposure time

10.2.4 Parameters

parameter	alias	default	min	max
uves_cal_mdark.stack_method	stack_method	median		

uves_cal_mdark.klow	klow	5.	0.	100.
uves_cal_mdark.khigh	khigh	5.	0.	100.
uves_cal_mdark.niter	niter	5	0	100
uves_cal_mdark.qc_dark.reg.num_x	qc_dark.reg.num_x	4	0	
uves_cal_mdark.qc_dark.reg.num_y	qc_dark.reg.num_y	4	0	
uves_cal_mdark.qc_dark.reg.box_sx	qc_dark.reg.box_sx	100	0	
uves_cal_mdark.qc_dark.reg.box_sy	qc_dark.reg.box_sy	100	0	
uves_cal_mdark.qc_dark.reg.border_x	qc_dark.reg.border_x	100	0	
uves_cal_mdark.qc_dark.reg.border_y	qc_dark.reg.border_y	100	0	
uves_cal_mdark.qc_dark.reg.when	qc_dark.reg.when	0	0	1

Note that the default parameters are robust.

10.3 uves_cal_predict

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

10.3.1 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_<CHIPID>	?	/path_pro/masterbias_<chipid>.fits
MASTER_FORM_<CHIPID>	?	/path_pro/master_form_<chipid>.fits

The master bias is an optional input since the measured arc line positions should not depend on master bias subtraction.

MASTER_FORM_<CHIPID> frames have been defined, only for standard settings and 1x1 binning, at the start of operations and for the RED arm detector upgrade, and are used for quality control.

10.3.2 Output

default recipe file name	format	PRO.CATG	short description
lineguesstable_<chipid>.fits	table	LINE_GUESS_TAB_<CHIPID>	Guess line table
orderguesstable_<chipid>.fits	table	ORDER_GUESS_TAB_<CHIPID>	Guess order table

The guess line table contains the following columns:

X Line position along x [pix]

Y	Same values as in Order column
PEAK	Line peak value [ADU]
Ident	Line catalog wavelength [Å]
YNEW	Predicted line y position [pix]
Order	Relative order number
WAVEC	Predicted wavelength of line peak [Å]
Aux	Product of wavelength and order number
XREG	X values after regression [pix]
Pixel	Local dispersion [Å/pix]
RORD	Order values after regression
XPRED	Predicted X line position [pix]
YPRED	Predicted Y line position [pix]
XDIF	Difference between measured and predicted X line position [pix]
YDIF	Difference between measured and predicted Y line position [pix]
SELPLOT	Selection column (to know which values where used after clipping of outliers)

The guess order table contains the following columns:

ABS_ORDER	Absolute order number
ORDER	Relative order number
X	Position along x [pix]
Y	Position along y [pix]
YFIT	Predicted order y position [pix]
RESIDUAL	Residual (Y-YFIT) [pix]

10.3.3 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 [pix]
QC.MODEL.DIFFXAVG	Mean difference of predicted and measured line x positions [pix]
QC.MODEL.DIFFXMED	Median difference of predicted and measured line x positions [pix]
QC.MODEL.DIFFYRMS	RMS difference of predicted and measured line y positions [pix]
QC.MODEL.DIFFYAVG	Mean difference of predicted and measured line y positions [pix]
QC.MODEL.DIFFYMED	Median difference of predicted and measured line y positions [pix]
QC.MODEL.WLENMIN	Minimum predicted lines wavelength [Å]
QC.MODEL.WLENMAX	Maximum predicted lines wavelength [Å]
QC.MODEL.ORDMIN	Minimum predicted absolute order
QC.MODEL.ORDMAX	Maximum predicted absolute order
QC.WLENMIN	Minimum wavelength of spectral format [Å]
QC.WLENMAX	Maximum wavelength of spectral format [Å]
QC.ORDMIN	Minimum relative order
QC.ORDMAX	Maximum relative order

10.3.4 Parameters

parameter	alias	default	min	max
uves_cal_predict.mbox_x	mbox_x	40	10	100
uves_cal_predict.mbox_y	mbox_y	40	10	100
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.def_pol1	def_pol1	4	0	
uves_cal_predict.def_pol2	def_pol2	5	0	
uves_cal_predict.kappa	kappa	4.5	0.	
uves_cal_predict.tol	tol	2.0	0.	

Note that **mbox_x, mbox_y, trans_x, trans_x, ccd_rot_angle_off** are parameters that may need to be adjusted (see Sec. 11.2.3). Other parameter values are robust.

10.4 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.

10.4.1 Input

frame tag/category	nr	filename example
ORDER_FLAT_<ARM>	1	/path_raw/uves_order_flat_<arm>.fits
ORDER_GUESS_TAB_<CHIPID>	?!	/path_pro/orderguesstable_<chipid>.fits
MASTER_BIAS_<CHIPID>	?	/path_pro/masterbias_<chipid>.fits

The input order guess table is recommended in order to ensure that the recipe detects the same orders as those predicted by the recipe uves_cal_predict, independently from the order flat frame illumination.

The master bias is an optional input since the measured order trace position should not depend on master bias subtraction.

10.4.2 Output

default recipe file name	format	PRO.CATG	short description
ordertable_<chipid>.fits	table	ORDER_TABLE_<CHIPID>	order table

The output table contains the columns:

Order	Relative order number
X	Position along x [pix]
Y	Order line centroid location [pix]
dY	Uncertainty on Y determination [pix]
Residual_Square	Squared residual of the initial 1D linear fit of the order [pix^2]
OrderRMS	Root mean squared residual of initial 1D linear fit of the order
OrderSlope	Slope of order
Yfit	The fitted order location [pix]
dYfit_Square	Variance of Yfit [pix^2]
Residual	(Y - Yfit) [pix]

10.4.3 Quality control

QC.ORD.RESIDMIN	Min residuals in order table [pix]
QC.ORD.RESIDMAX	Max residuals in order table [pix]
QC.ORD.RESIDAVG	Mean residuals in order table [pix]
QC.ORD.RESIDRMS	RMS residuals in order table [pix]
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

In this table, the residuals refer to 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 11.11

10.4.4 Parameters

parameter	alias	default	min	max
uves_cal_orderpos.preproc.use_guess_tab	use_guess_tab	1	0	2
uves_cal_orderpos.preproc.radx	radx	2	0	
uves_cal_orderpos.preproc.rady	rady	1	0	
uves_cal_orderpos.preproc.mmethod	mmethod	median		
uves_cal_orderpos.preproc.backsubgrid	backsubgrid	50	10	
uves_cal_orderpos.preproc.backsubradiusy	backsubradiusy	2	0	
uves_cal_orderpos.preproc.backsubkappa	backsubkappa	4.	0.	
uves_cal_orderpos.preproc.backsubdegx	backsubdegx	2	1	

uves_cal_orderpos.preproc.backsubdegy	backsubdegy	2	1	
uves_cal_orderpos.hough.samplewidth	samplewidth	50	1	
uves_cal_orderpos.hough.minslope	minslope	0.0	0.0	maxslope
uves_cal_orderpos.hough.maxslope	maxslope	0.2	minslope	
uves_cal_orderpos.hough.sloperes	sloperes	120	1	
uves_cal_orderpos.hough.pthres	pthres	0.2	0.	1.
uves_cal_orderpos.trace.tracestep	tracestep	10	1	
uves_cal_orderpos.trace.minthresh	minthresh	0.2	0.	1.
uves_cal_orderpos.trace.maxgap	maxgap	0.2	0.	1.
uves_cal_orderpos.reject.maxrms	maxrms	100.0	0.	
uves_cal_orderpos.reject.defpol1	defpol1	-1	-1	
uves_cal_orderpos.reject.defpol2	defpol2	-1	-1	
uves_cal_orderpos.reject.kappa	kappa	4.	0.	

Note that the default parameters are robust.

10.5 uves_cal_mflat

The recipe uves_cal_mflat creates the master flat frame.

10.5.1 Input

frame tag/category	nr	filename example
FLAT_<ARM>	+	/path_raw/uves_flat_<arm>.fits
MASTER_BIAS_<CHIPID>	?!	/path_pro/masterbias_<chipid>.fits
MASTER_DARK_<CHIPID>	?	/path_pro/masterdark_<chipid>.fits
ORDER_TABLE_<CHIPID>	1	/path_pro/ordertable_<chipid>.fits

The optional input master dark is subtracted if provided. The UVES dark count level is usually negligible.

10.5.2 Output

default recipe file name	format	PRO.CATG	short description
masterflat_<chipid>.fits	image	MASTER_FLAT_<CHIPID>	master flat frame
masterflat_bkg_<chipid>.fits	image	BKG_FLAT_<CHIPID>	background of master flat frame

10.5.3 Quality control

The recipe computes the following QC parameters:

QC.DELTA.TEMP4	Difference INS.TEMP4.MEAN to reference
QC.DELTA.YSHIFT	Difference Y-shift to reference
QC.SIGMA.PH	Photon noise
QC.SIGMA.DX	Noise from DX derivative
QC.SIGMA.DY	Noise from DY derivative
QC.SIGNAL	Mean value across subwindow

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	min	max
uves_cal_mflat.norm_method	norm_method	explevel		
uves_cal_mflat.backsub.mmethod	backsub.mmethod	median		
uves_cal_mflat.backsub.npoints	backsub.npoints	82	0	
uves_cal_mflat.backsub.radiusy	backsub.radiusy	2	0	
uves_cal_mflat.backsub.sdegree	backsub.sdegree	1	0	
uves_cal_mflat.backsub.smoothx	backsub.smoothx	-1.0		
uves_cal_mflat.backsub.smoothy	backsub.smoothy	-1.0		

Note that the default parameters are robust.

10.6 uves_cal_mflat_combine

The recipe uves_cal_mflat_combine combines the master flat frame obtained with the ^{2}D lamp (BLUE arm 346 setting) with a normal master flat of the same setting. The two flats are merged along a trace passing through the middle between the order 145 and order 146. To ensure a smooth intensity change along the merging trace, the ^{2}D lamp flat intensity needs to be rescaled appropriately. The corresponding scale factor is computed as the ratio between the median flux measured in a square box of 5x5 pixels at the center of the order 145 of the normal flat, and the same quantity on the order 146 of the ^{2}D lamp flat.

10.6.1 Input

frame tag/category	nr	filename example
MASTER_FLAT_BLUE	?	/path_pro/masterflat_blue.fits
MASTER_DFLAT_BLUE	?	/path_pro/masterdflat_blue.fits
ORDER_TABLE_BLUE	1	/path_pro/ordertable_blue.fits

At least one of the master flat input frames must be provided. If only one frame is provided the product corresponds to the input frame.

10.6.2 Output

default recipe file name	format	PRO.CATG	short description
masterflat_blue.fits	image	MASTER_FLAT_BLUE	master (combined) flat frame

10.6.3 Quality control

No quality control.

10.6.4 Parameters

parameter	alias	default	min	max
uves.order_threshold	order_threshold	7	5	9

Note that the default parameters are robust.

10.7 uves_cal_mkmaster

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

10.8 uves_cal_wavecal

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

10.8.1 Input

frame tag/category	nr	filename example
ARC_LAMP_<ARM>	1	/path_raw/uves_arc_lamp_<arm>.fits
ORDER_TABLE_<CHIPID>	1	/path_pro/ordertable_<chipid>.fits
LINE_GUESS_TAB_<CHIPID>	1	/path_pro/lineguesstable_<chipid>.fits
LINE_REFER_TABLE	1	/path_ref/thargood_3.fits
LINE_INTMON_TABLE	?	/path_ref/line_intmon.fits
MASTER_BIAS_<CHIPID>	?	/path_pro/masterbias_<chipid>.fits
MASTER_DARK_<CHIPID>	?	/path_pro/masterdark_<chipid>.fits
MASTER_FLAT_<CHIPID>	?!	/path_pro/masterflat_<chipid>.fits

WEIGHTS_<CHIPID>	?	/path_pro/weights_<chipid>.fits

The input master flat is recommended in order to obtain a more accurate line peak detection, particularly for the arc lines that lie on the steep part of the blaze slope.

The master bias is an optional input since the measured arc line positions should not depend on master bias subtraction.

The optional input WEIGHTS_<CHIPID> is a map of pixel weights obtained after optimal extraction of the science spectrum. A user requiring a very accurate wavelength calibration may first reduce the data without this input, using optimal extraction to extract the object spectrum, and in a second iteration, use the weights from the optimal extraction for wavelength calibration and repeat the science reduction with the optimized line list.

10.8.2 Output

default recipe file name	format	PRO.CATG	short description
linetable_<chipid>.fits	table	LINE_TABLE_<CHIPID>	line table

The first extension output line table, LINE_TABLE_<CHIPID>, contains the columns:

X	Line X position [pix]
dX	Uncertainty on X [pix]
XWidth	Line sigma [pix]
Y	Relative order number of detected line
Peak	Line peak counts [ADU]
Background	Fitted background of detected line [ADU]
Slope	Linear background slope of detected line [ADU/pix]
Order	Absolute order number of detected line
AbsOrder	Absolute order number of detected line
Ynew	Line Y position [pix]
WaveC	Line wavelength (computed using the fitted dispersion relation) [Å]
dLambdaC	Uncertainty on WaveC [Å]
Pixel	Local dispersion [Å/pix]
Residual	Line residual [Å]
Residual_pix	Line residual [pix]
Lambda_candidate	Wavelength of nearest line in catalogue [Å]
dLambda_candidate	Uncertainty on Lambda_candidate [Å]
dLambda_cat_sq	Squared wavelength residual from the nearest catalogue line [$Å^2$]
dLambda_nn_sq	Squared wavelength residual from nearest neighbour multiplied by ALPHA [$Å^2$]
Ident	Catalogue wavelength associated with this emission line, or NULL if this line was not identified.
dIdent	Uncertainty on catalogue line 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	Line peak counts scaled to unit exposure time (this column is present only if a LINE_INTMON_TABLE is provided)

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

If there is more than one extraction window (default is **nwindows=**3), then the results of each extraction and wavelength calibration are stored in subsequent extensions of the same FITS table. 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 header keywords as "HISTORY OFFSET -7.5". The corresponding window number (counting from 1) is stored in header keywords as "HISTORY WINDOW 3"

The polynomials are stored in table extensions as follows:

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 first six rows of the table define a linear transformation of the dependent and independent variables. For example the table shown above 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.

10.8.3 Quality control

The recipe computes the following quality control parameters:

QC.TRACE0.WINi.NLINDETj	ThAr lamp detected lines on trace 0, extraction window i, iteration j
QC.TRACE0.WINi.NLINDET.NITERS	Number of iterations on trace 0, extraction window i
QC.TRACE0.WINi.NLINIDj	ThAr lamp identified lines on trace 0, extraction window i, iteration j
QC.TRACE0.WINi.NLINID.NITERS	Number of iterations on trace 0, extraction window i
QC.FWHMAVG	Average FWHM of fitted lines on TRACE0 WIN2 [pix]
QC.FWHMRMS	Standard deviation of FWHM of fitted lines on TRACE0 WIN2 [pix]
QC.FWHMMED	Median FWHM of fitted lines on TRACE0 WIN2 [pix]
QC.RESOLAVG	Average resolving power of fitted lines on TRACE0 WIN2
QC.RESOLRMS	Standard deviation of the resolving power of fitted lines on TRACE0 WIN2
QC.RESOLMED	Median resolving power of fitted lines on TRACE0 WIN2
QC.LINE.RESIDAVG	Mean line residual on TRACE0 WIN2 [pm]
QC.LINE.RESIDRMS	Standard deviation of line residuals on TRACE0 WIN2 [pm]
QC.LINE.RESIDRMS.WLU	Line identification RMS on TRACE0 WIN2 [Å]
QC.LINE.RESIDRMS.PIX	Line identification RMS on TRACE0 WIN2 [pix]
QC.LINE.RESIDRMS.SPEED	Line identification RMS on TRACE0 WIN2 [m/s]
QC.LINE.IDCHI2	Reduced chi^2 of line identification on TRACE0 WIN2
QC.LINE.HALFBRIG	Fraction of lines with intensity grater than half of the brightest one on TRACE0 WIN2
QC.WLENMIN	Minimum wavelength of detected orders on TRACE0 WIN2 [nm]
QC.WLENMAX	Maximum wavelength of detected orders on TRACE0 WIN2 [nm]
QC.ORDMIN	Minimum order number expected on TRACE0 WIN2
QC.ORDMAX	Maximum order number expected on TRACE0 WIN2
QC.ORDMIN.DETECTED	minimum order number detected on TRACE0 WIN2
QC.ORDMAX.DETECTED	maximum order number detected on TRACE0 WIN2
QC.NLINTOT	Total number of lines found on the input arc frame
QC.NLINSEL	Number of identified lines on TRACE0 WIN2
QC.NLINSOL	Number of identified and fitted lines on TRACE0 WIN2
QC.LINE WAVEERR	Wavelength error on trace 0 [Å]
QC.LINE SYSERR	Wavelength systematic error on trace 0 [Å]
QC.NLINRES1	Number of identified and fitted lines with residuals < 1 Åon TRACE0 WIN2
QC.INTAVG	Average intensity of line list on TRACE0 WIN2
QC.NLININT	No of lines to measure INTAVG on TRACE0 WIN2

The line FWHMs and the corresponding resolving power are measured along the dispersion direction (see also Figure 11.13). The residuals correspond to the differences between the solution from the fitted polynomial model and the catalogue wavelengths for the lines identified on the arc lamp frame (see also Figure 11.12).

10.8.4 Parameters

parameter	alias	default	min	max

uves_cal_wavecal.nwindows	nwindows	3	1	
uves_cal_wavecal.length	length	-1.	-2.	
uves_cal_wavecal.offset	offset	0.	-25.	25.
uves_cal_wavecal.extract.method	extract.method	average		
uves_cal_wavecal.extract.kappa	extract.kappa	10.0	0.	100.
uves_cal_wavecal.extract.chunk	extract.chunk	32	1	
uves_cal_wavecal.extract.profile	extract.profile	auto		
uves_cal_wavecal.extract.skymethod	extract.skymethod	optimal		
uves_cal_wavecal.extract.oversample	extract.oversample	-1	-2	
uves_cal_wavecal.extract.best	extract.best	TRUE		
uves_cal_wavecal.search.range	range	8	1	
uves_cal_wavecal.search.minlines	minlines	0	0	maxlines
uves_cal_wavecal.search.maxlines	maxlines	0	minlines	
uves_cal_wavecal.first.shiftmax	shiftmax	10.	0.	
uves_cal_wavecal.first.shiftstep	shiftstep	0.1	0.	
uves_cal_wavecal.first.shifttoler	shifttoler	0.05	0.	
uves_cal_wavecal.identify.alpha	alpha	0.1	0.	
uves_cal_wavecal.identify.maxerror	maxerror	20.0	0.	
uves_cal_wavecal.identify.degree	degree	4	-2	
uves_cal_wavecal.calibrate.tolerance	tolerance	0.6		
uves_cal_wavecal.calibrate.kappa	kappa	4.0	0.	100.
uves_cal_wavecal.rebin.wavestep	rebin.wavestep	-1.0	-1.	
uves_cal_wavecal.rebin.wavestep_redu	rebin.wavestep_redu	-1.0	-1.	
uves_cal_wavecal.rebin.scale	rebin.scale	FALSE		

Please be aware that setting large values for the parameter wavestep (or wavestep_redu) will cause the recipe to fail.

Note that the parameter **tolerance** is a parameter that may need to be adjusted in order to obtain the best wavelength solution (it should be kept small, for example equal to 0.07 in order to take advantage of the good reference line list table, thargood_3.fits).

The **maxlines** and **minlines** parameter values are set by default to 0, meaning that a hard-coded default value (1100 for BLUE/REDL chips; 1000 for REDU chip) is used for the minimum number of lines to detect, and (1600 for BLUE/REDL chip; 1400 for REDU chip) is used for the maximum number of lines to detect. The user may change these defaults, choosing relatively large number (to search at least five lines per order).

Other parameters are robust.

Note that **rebin.wavestep** controls the wavelength binning setting for BLUE and REDL arms while **rebin.wavestep_redu** controls the wavelength binning setting for the REDU arm.

10.8.5 Object-weighted extraction

Increased accuracy of the dispersion relation may be obtained by extracting the arc lamp frame using the same weights as those used for the science object optimal extraction. The necessary steps in this procedure are as

follows:

- Run uves_cal_wavecal to produce the table LINE_TABLE_<CHIPID> (this product is used only to bootstrap the next step, so no particular accuracy is required).
- Run uves_obs_scired with the --debug option set to true. This will create the FITS image file weights_<chipid>.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_<chipid>.fits file, which must be classified as WEIGHTS_<CHIPID>. This will generate the final LINE_TABLE_<CHIPID> containing the dispersion solution based on the weighted arc lamp extraction. [11]
- Re-run uves_obs_scired using the LINE_TABLE_<CHIPID> generated in the previous step.

10.9 uves_cal_response

10.9.1 Input

frame tag/category	nr	filename example
STANDARD_<ARM>	1	/path_raw/uves_standard_<chipid>.fits
ORDER_TABLE_<CHIPID>	1	/path_pro/ordertable_<chipid>.fits
LINE_TABLE_<CHIPID>	1	/path_pro/linetable_<chipid>.fits
MASTER_BIAS_<CHIPID>	?!	/path_pro/masterbias_<chipid>.fits
MASTER_DARK_<CHIPID>	?	/path_pro/masterdark_<chipid>.fits
MASTER_FLAT_<CHIPID>	1	/path_pro/masterflat_<chipid>.fits
FLUX_STD_TABLE	1	/path_ref/flxstd.fits
EXTCOEFF_TABLE	1	/path_ref/uves_paranal_extinct_model.fits
RESP_FIT_POINTS_CATALOG	1	/path_ref/RESP_FIT_POINTS_CATALOG.fits

The FLUX_STD_TABLE, is the catalog of flux std stars. The EXTCOEFF_TABLE, is the atmospheric extinction correction table. The RESP_FIT_POINTS_CATALOG, is the catalog indicating points to fit the response.

10.9.2 Output

default recipe file name	format	PRO.CATG	short description
response_fine_<chipid>.fits	table	INSTR_RESPONSE_FINE_<CHIPID>	instrument response (three extensions for: smoothed response, raw response, and fit points)
red_std_<chipid>.fits	1d (wav) image	RED_STD_<CHIPID>	extracted merged std star spectrum
order_extract_qc_<chipid>.fits	table	ORDER_EXTRACT_QC_<CHIPID>	table of relevant qc parameters as a function of order

[11] For simplicity and consistency with the usual products of uves_cal_wavecal, this LINE_TABLE_<CHIPID> contains the same dispersion solution three times.

efficiency_<chipid>.fits	table	EFFICIENCY_TABLE_<CHIPID>	instrument efficiency
bkg_std_<chipid>.fits	2d (pix-pix) image	BKG_STD_<CHIPID>	sky background of std star

The instrument efficiency table contains the following columns:

Wave	Wavelength at blaze centre [Å]
Eff	Efficiency at blaze centre
Binsize	Size of bin [Å]
Order	Relative order number

10.9.3 Quality control

The pipeline measures the following quality control parameters:

QC.ORDi.OBJ.SN	Average S/N at centre of blaze
QC.ORDi.OBJ.POS	Average OBJ POS at centre of blaze [pix]
QC.ORDi.OBJ.FWHM	Average spatial FWHM for the order [pix]
QC.ORDi.OBJ.RPLPAR	Average relative ripple amplitude
QC.EX.NORD	Number of orders extracted
QC.EX.XSIZE	Input image width [pix]
QC.EX.YSIZE	Extraction slit length [pix]
QC.VRAD.BARYCOR	Barycentric radial velocity correction [km/s]
QC.VRAD.HELICOR	Heliocentric radial velocity correction [km/s]
QC.MEAN.RAW	Mean level of raw file
QC.MEAN.RESP	Mean level of response curve
QC.LAMBDA.MAX	Maximum wavelength [A]
QC.NORDER	Number of orders
QC.WLEN	Central blaze wavelength [A]
QC.EFF.MAX	Efficiency value at WLEN

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

Additional QC parameters are logged in the instrument efficiency table:

QC.BLAZEFFi	Efficiency at blaze maximum
QC.BLAZWLENi	Wavelength at blaze maximum [Å]

See Figure 11.15 for typical plots of the instrument response and the instrument efficiency curves.

10.9.4 Parameters

parameter	alias	default	min	max

uves_cal_response.reduce.backsub.mmethod	reduce.backsub.mmethod	median		
uves_cal_response.reduce.backsub.npoints	reduce.backsub.npoints	82	0	
uves_cal_response.reduce.backsub.radiusy	reduce.backsub.radiusy	2	1	
uves_cal_response.reduce.backsub.sdegree	reduce.backsub.sdegree	1	0	
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	-1.	100.
uves_cal_response.reduce.extract.chunk	reduce.extract.chunk	32	1	
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	-1	-2	
uves_cal_response.reduce.extract.best	reduce.extract.best	TRUE		
uves_cal_response.reduce.slitlength	reduce.slitlength	-1.	-2.	
uves_cal_response.reduce.skysub	reduce.skysub	TRUE		
uves_cal_response.reduce.objoffset	reduce.objoffset	0.		
uves_cal_response.reduce.objslit	reduce.objslit	-1.	-2.	
uves_cal_response.reduce.tiltcorr	reduce.tiltcorr	TRUE		
uves_cal_response.reduce.ffmethod	reduce.ffmethod	extract		
uves_cal_response.reduce.rebin.wavestep	reduce.rebin.wavestep	-1.		
uves_cal_response.rebin.wavestep_redu	rebin.wavestep_redu	-1.0	-1.	
uves_cal_response.reduce.rebin.scale	reduce.rebin.scale	FALSE		
uves_cal_response.reduce.merge	reduce.merge	optimal		
uves_cal_response.reduce.merge_delt1	reduce.merge_delt1	0.	0.	20.
uves_cal_response.reduce.merge_delt2	reduce.merge_delt2	0.	0.	20.
uves_cal_response.efficiency.reduce.extract.method	efficiency.reduce.extract.method	linear		
uves_cal_response.efficiency.reduce.ffmethod	efficiency.reduce.ffmethod	no		
uves_cal_response.efficiency.reduce.merge	efficiency.reduce.merge	sum		
uves_cal_response.efficiency.paccuracy	paccuracy	60.		

Note that **reduce.extract.method, reduce.extract.kappa, reduce.extract.profile**, **reduce.extract.oversample** are parameters that may need to be adjusted in order to obtain the best standard star spectrum extraction. Other parameters are robust.

Note that **rebin.wavestep** controls the wavelength binning setting for BLUE and REDL arms while **rebin.wavestep_redu** controls the wavelength binning setting for the REDU arm.

Note that **reduce.rebin.scale** should be set to **TRUE** if **reduce.ffmethod** is set to **no** (and only then), because flat fielding already ensures flux conservation.

10.10 uves_obs_scired

This recipe reduces a science frame and extracts the spectrum.

10.10.1 Input

frame tag/category	nr	filename example
SCIENCE_<ARM>	1	/path_raw/uves_science_<chipid>.fits
ORDER_TABLE_<CHIPID>	1	/path_pro/ordertable_<chipid>.fits

LINE_TABLE_<CHIPID>	1	/path_pro/linetable_<chipid>.fits
MASTER_BIAS_<CHIPID>	?!	/path_pro/masterbias_<chipid>.fits
MASTER_DARK_<CHIPID>	?	/path_pro/masterdark_<chipid>.fits
MASTER_FLAT_<CHIPID>	1	/path_pro/masterflat_<chipid>.fits
EXTCOEFF_TABLE	?	/path_ref/uves_paranal_extinct_model.fits
INSTR_RESPONSE_FINE_<CHIPID>	?	/path_pro/response_fine_<chipid>.fits
MASTER_RESPONSE_<CHIPID>	?	/path_raw/master_response_<chipid>.fits

If the atmospheric extinction table and the instrument response are provided, then this recipe also generates a flux calibrated spectrum and its uncertainty.

Note: the user should provide either the INSTR_RESPONSE_FINE_<CHIPID> or the MASTER_RESPONSE_<CHIPID> (that is generated by the ESO Quality Control Group UVES instrument responsible). If the user provides both, then the MASTER_RESPONSE_<CHIPID> is used. In general the MASTER_RESPONSE_<CHIPID> table gives better results than the INSTR_RESPONSE_FINE_<CHIPID> table derived from a single observation.

10.10.2 Output

default recipe file name	format	PRO.CATG	short description
red_science_<chipid>	1d (wav) image	RED_SCIENCE_<CHIPID>	extracted, flatfielded, wavelength calibrated, merged, sky subtracted science spectrum
merged_science_<chipid>	1d (wav) image	MERGED_SCIENCE_<CHIPID>	extracted, flatfielded, wavelength calibrated, merged science spectrum
resampled_science_<chipid>	2d (wav-order) image	WCALIB_SCIENCE_<CHIPID>	extracted, flatfielded, wavelength calibrated, science frame
resampled_error_science_<chipid>	2d (wav-order) image	ERRORBAR_WCALIB_SCIENCE_<CHIPID>	uncertainty on WCALIB_SCIENCE_<CHIPID>
order_extract_qc_<chipid>	table	ORDER_EXTRACT_QC_<CHIPID>	table of relevant qc parameters as a function of order
resampled_ff_science_<chipid>	2d (wav-order) image	WCALIB_FF_SCIENCE_<CHIPID>	resampled, (if reduce.ffmethod='pixel' also flat fielded), wavelength calibrated science frame[12]
resampled_mflat_<chipid>	2d (wav-pix) image	WCALIB_FLAT_OBJ_<CHIPID>	resampled master flat field frame
error_red_science_<chipid>	1d (wav) image	ERRORBAR_SCIENCE_<CHIPID>	uncertainty on RED_SCIENCE_<CHIPID>
variance_ff_science_<chipid>	1d (wav) image	VARIANCE_SCIENCE_<CHIPID>	variance on WCALIB_FF_SCIENCE_<CHIPID>
background_<chipid>	2d (pix-pix) image	BKG_SCI_<CHIPID>	background of science frame
ordertrace_<chipid>	table	ORDER_TRACE_<CHIPID>	order trace table
cr_mask_<chipid>	table	CRMASK_<CHIPID>	cosmic ray mask table
merged_sky_<chipid>	1d (wav) image	MERGED_SKY_<CHIPID>	merged sky spectrum
fluxcal_science_<chipid>	1d (wav) image	FLUXCAL_SCIENCE_<CHIPID>	flux calibrated version of RED_SCIENCE_<CHIPID>
fluxcal_error_science_<chipid>	1d (wav) image	FLUXCAL_ERRORBAR_SCIENCE_<CHIPID>	uncertainty on FLUXCAL_SCIENCE_<CHIPID>
wave_map_<chipid>	2d (pip-pix) image	WAVE_MAP_<CHIPID>	wavelength map[13]

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

10.10.3 Quality control

The recipe computes the following quality control parameters for the extracted science spectrum and stored in its FITS header.

[12] This provides the spectrum orders. The wavelength information is stored in the header keywords WSTARTi,WENDi, with i being the number of the order

[13] To be compared to the product b_<chipid>.fits which can be obtained by setting -debug=true

QC.ORDi.OBJ.SN	Average S/N at centre of blaze
QC.ORDi.OBJ.POS	Average OBJ POS at centre of blaze [pix]
QC.ORDi.OBJ.FWHM	Average spatial FWHM for the order [pix]
QC.ORDi.OBJ.RPLPAR	Average relative ripple amplitude
QC.EX.NORD	Number of orders extracted
QC.EX.XSIZE	Input image width [pix]
QC.EX.YSIZE	Extraction slit length [pix]
QC.VRAD.BARYCOR	Barycentric radial velocity correction [km/s]
QC.VRAD.HELICOR	Heliocentric radial velocity correction [km/s]
QC.MEAN.RAW	Mean level of (first) raw file
QC.MEAN.RED	Mean level of reduced file
QC.N.SAT	Number of saturated pixels in (first) raw frame
QC.SNR	Mean S/N from QC.ORDi.OBJ.SN
QC.MEAN.SNR	Mean S/N from ratio signal/errorbar
QC.MEAN.RPL	Mean ripple parameter from QC.ORDi.OBJ.RPLPAR
QC.MEAN.POS	Mean signal position in slit from QC.ORDi.OBJ.POS [pixels]
QC.MEAN.FWHM	Mean signal width from QC.ORDi.OBJ.FW [pixels]
QC.WLEN.START	Start wavelength [A]
QC.WLEN.END	End wavelength [A]
QC.STEP	Step size [A]
QC.AIRM	Mean airmass during (first) exposure
QC.FWHM	Mean seeing during (first) exposure
QC.REL.POS	Single only: relative position of maximum (0.5: slit centre)
QC.MED.REDUCED	Median value of reduced file
QC.ADD.COMP	Additional components (absorption cell, image slicers)
QC.PROC.SCHEME	Processing scheme
IS_FLUXD	Is flux calibrated spectrum available?
VM_SM	Visitor Mode, Service Mode, or unknown (NO)
QC_FLAG	Quality flags

where i is the order number.

The parameters SN, POS, FWHM, RPLPAR give, order by order, an indication of the measured object signal to noise ratio, position along the slit at the centre of the order, spatial FWHM, and extraction quality parameter (RPLPAR is a measure of the amplitude of systematic ripples in the extracted spectrum). If the POS parameter reaches values close to zero or the slit length, probably due to a high value of the airmass, then the object may fall outside the extraction slit. In these cases we suggest the use of an order trace obtained from a standard star observation with a similar airmass. If the object spatial FWHM is comparable to the extraction slit length, then we recommend that the user employs 2D extraction. If the ripple parameter is much higher than 5, then we recommend that the user increases the value of the parameter **reduce.extract.oversample** (which will proportionally increase the execution time). If this does not work appropriately, then the user may opt to employ the linear or average extraction methods.

In all science products with a wavelength scale are present also the following keywords:

LAMRMS	RMS of wavelength solution [Ang]
LAMNLIN	No lines used in wavelength solution
LAMERR	Wavelength uncertainty [Ang]
LAMSYS	Typical systematic wavelength error [Ang]

10.10.4 Parameters

parameter	alias	default	min	max
uves_obs_scired.clean_traps	clean_traps	FALSE		
uves_obs_scired.reduce.backsub.mmethod	reduce.backsub.mmethod	median		
uves_obs_scired.reduce.backsub.npoints	reduce.backsub.npoints	82	0	
uves_obs_scired.reduce.backsub.radiusy	reduce.backsub.radiusy	2	1	
uves_obs_scired.reduce.backsub.sdegree	reduce.backsub.sdegree	1	0	
uves_obs_scired.reduce.backsub.smoothx	reduce.backsub.smoothx	-1.		
uves_obs_scired.reduce.backsub.smoothy	reduce.backsub.smoothy	-1.		
uves_obs_scired.reduce.extract.method	reduce.extract.method	optimal		
uves_obs_scired.reduce.extract.kappa	reduce.extract.kappa	10.	-1.	100
uves_obs_scired.reduce.extract.chunk	reduce.extract.chunk	32	1	
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	-1	-2	
uves_obs_scired.reduce.extract.best	reduce.extract.best	TRUE		
uves_obs_scired.reduce.slitlength	reduce.slitlength	-1.	-2.	
uves_obs_scired.reduce.skysub	reduce.skysub	TRUE		
uves_obs_scired.reduce.objoffset	reduce.objoffset	0.		
uves_obs_scired.reduce.objslit	reduce.objslit	-1.	-2.	
uves_obs_scired.reduce.tiltcorr	reduce.tiltcorr	TRUE		
uves_obs_scired.reduce.ffmethod	reduce.ffmethod	extract		
uves_obs_scired.reduce.rebin.wavestep	reduce.rebin.wavestep	-1.0		
uves_obs_scired.rebin.wavestep_redu	rebin.wavestep_redu	-1.0	-1.	
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	0.	20.
uves_obs_scired.reduce.merge_delt2	reduce.merge_delt2	0.0	0.	20.
uves_obs_scired.velocity_correction	velocity_correction	none		

We provide the user with the following science data reduction hints:

- The parameter **reduce.extract.method** allows the user to set the extraction method. This must be appropriate to the observed object depending on its S/N ratio, if it is point-like or extended, if it was observed in SLIT, IS, or I2-Cell mode.
- In certain cases the reduced merged spectra may show evenly spaced spikes which correspond to merging problems at the order boundaries. In order to improve the extraction quality of these spectra, the user may

need to set proper values of the parameters **reduce.merge_delt1** and **reduce.merge_delt2** that control the wavelength extension of the contributions from the overlapping parts of adjacent orders. For example values of 10 and 2 respepectively for parameters **reduce.merge_delt1** and **reduce.merge_delt2** may generate better quality spectra.

- In case of extraction of objects that have multiple spectral traces along the slit the user may obtain proper results by setting appropriately the parameters **reduce.objoffset** and **reduce.objslit**. We remind the user to consult the on-line recipe help for a more detailed description of these parameters.

- (*Optional*) In case of data observed in the RED860 (or RED760) setting, in order to reduce effects from fringes, we recommend the user to set the parameter **reduce.ffmethod** to 'pixel', that performs a flat-fielding before extraction.

Note that **reduce.extract.method, reduce.extract.kappa, reduce.extract.profile**, **reduce.extract.oversample** are parameters that may need to be adjusted in order to obtain the best standard star spectrum extraction. Other parameters are robust.

Note that **rebin.wavestep** controls the wavelength binning setting for BLUE and REDL arms while **rebin.wavestep_redu** controls the wavelength binning setting for the REDU arm.

Note that **reduce.ffmethod** has to be the same for response and science.

Note that **reduce.rebin.scale** should be set to **TRUE** if **reduce.ffmethod** is set to **no** (and only then), because flat fielding already ensures flux conservation.

10.11 uves_obs_redchain

This recipe performs the full UVES data reduction.

10.11.1 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>	?	/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	?	/path_ref/flxstd.fits
EXTCOEFF_TABLE	?	/path_ref/uves_paranal_extinct_model.fits

10.11.2 Output

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

10.11.3 Quality control

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

10.11.4 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 the execution of the last step on/off (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 implements full error propagation by using the error propagation formula and making the usual assumption about Gaussian error bars. The initial variance σ_i^2 for the ith pixel in units of e^{-2} is defined by:

$$\sigma_i^2 = G^2\sigma_{\rm RON}^2 + (1 - G^2)/12 + G(X_i - B_i) \quad (1)$$

where $\sigma_{\rm RON}$ is the readout noise (ADU), G is the gain (e^-/ADU), X_i is the raw pixel counts (ADU), and B_i is the bias level (ADU). Note that the term $(1 - G^2)/12$ represents the discretisation noise in converting from electrons to ADU, and if G happens to be greater than 1, then the term is ignored.

The values of G and $\sigma_{\rm RON}$ are taken from the header keywords HIERARCH ESO DET OUT1 GAIN and HIERARCH ESO DET OUT1 RON, respectively.

11.1.2 Master frame creation using a kappa-sigma-clipped mean

A kappa-sigma-clipped mean may be used on a per-pixel basis to combine a set of frames into a master frame, thereby excluding outlier pixel values. The user may set the number of iterations (**niter**) and the kappa values controlling the low (**klow**) and high (**khigh**) count thresholds used to clip outliers. For a more robust procedure, the first iteration of the kappa-sigma-clipping employs the median of the counts at a given pixel.

11.1.3 Guess table generation

The purpose of this algorithm is to generate two tables, one with the physical model predicted X and Y positions on the detector for each of the reference spectral format table lines, and another table with the predicted free spectral range corresponding to a given instrument setting and and ambient conditions (temperature, pressure, atmospheric humidity).

The relevant user input parameters are read: **mbox_x,mbox_y,trans_x,trans_y**, **ech_angle_off**, **cd_angle_off**, **ccd_rot_angle_off**, **compute_regression_sw**.

The instrument configuration, CCD id, Echelle/FIBER mode (and in this case the plate number setting), the detector X and Y binning, the ambient pressure, the humidity (this is a hard-coded value corresponding to the standard Paranal operational conditions), the slit width, slit length, central wavelength, cross disperser ID, camera temperature, and frame chip ID, are read from the input raw frame FITS header and logged to the recipe output.

Additional X and Y and CCD rotation angle corrections are applied to the physical model to obtain a better match to the actual spectral format, depending on wavelength and chip (and detector history) settings, which have been obtained during commissioning.

The refractive index corresponding to a given wavelength, temperature and pressure is computed.

In case of FIBER mode observations, to take into account of the shift of the calibration fibre corresponding to a given plate from the instrument slit central trace, corrective X and Y translation parameters are computed as a function of the observing plate, detector chip and grating wavelength settings.

The X, Y, echelle, cross-disperser and CCD angles are corrected in the case of new RED detector data.

The CCD rotation angles and blaze incidence angles onto the echelle grating and cross disperser offsets are set.

The instrument physical model X and Y user defined offsets (**trans_x,trans_y**) are applied.

Then for each wavelength specified by the reference arc line table the following quantities are calculated: the order within which a corresponding photon would arrive on the detector, the emerging angles from the echelle and the cross disperser gratings, the focal length, the blaze function, and the X and Y pixel positions corresponding to the nominal order and the two adjacent ones. If the X and Y positions fall on the detector, then they are written to the result table. The line FWHM (in pixels and Å) and the corresponding spectral resolution are computed.

If **compute_regression_sw** is set to TRUE, then a linear regression of the X,Y guess positions is performed.

Finally the output table will contain information on wavelength, order, X and Y model guesses, detector bin size, line width in pixel and Å, resolution, line length in pixels, blaze value.

Finally the Free Spectral Range is determined.

11.1.4 Reference arc line position measure

This algorithm measures the raw XY pixel positions and widths of the Th-Ar lines on a ThAr arc echelle exposure. It also computes the differences with the predicted positions.

The input parameters **mbox_x, mbox_y** and detector X and Y binning are read. The 2D Gaussian fit search rectangle is set, correcting for the input frame bin size. Then, for each line in the model spectral format, a 2D Gaussian fit is performed within the rectangular fit box centered at the predicted line position, ignoring fit boxes outside of the detector area.

Finally a 2D polynomial fit of degree 2 in X and Y is performed to the X and Y fitted line positions via regression.

11.1.5 Guess line and order tables generation

This procedure creates the order definitions and wavelength calibration from the wavelength projection map, and saves the data in order and line tables.

The input parameters **kappa** and **tol** are read.

A fit of the arc line positions as measured on the spectral format frame are used to trace the orders via a bivariate polynomial.

A kappa-sigma-clip is applied to the fit residuals where the kappa of the sigma clip is defined by the parameter **kappa**. In the case that the sigma clip threshold is greater than the value of the parameter **tol** then the sigma-clip-threshold is set to the value of **tol** to ensure reasonable data clipping.

Points for which **kappa** times the mean quadratic error associated to each Y position after 2D polynomial regression (**def_pol1, def_pol2**) is greater than **tol** are clipped out.

A 1D polynomial fit of degree 3 of the AUX as function of X is performed.

Finally a 2D polynomial fit of the order as function of the guess X and Y positions is performed.

11.1.6 Spectral format stability check

A reference spectral format frame is a frame used to optimize the UVES physical model parameters, and it will have been taken when UVES operations started, or after a detector upgrade. If the user provides an input reference spectral format frame, then the uves_cal_predict recipe computes similar tables for both the current and reference format-check frames, and then correlates the results to obtain a measure of the global shift in X and Y of the current frame with respect to the reference frame.

11.1.7 Combination of blue master ^{2}D and normal flat frames

In order to improve extraction quality in of science data taken in the 346 setting the user may combine the master flats obtained reducing frame acquired with the ^{2}D lamp with the one obtained with the normal lamp.

To properly combine the frames the pipeline uses all but the last seven traced orders from the standard lamp master flat with the last **order_threshold** traced orders from the ^{2}D master flat, rescaled in intensity.

The intensity scaling factor is computed as the ratio of the two median values measured on the normal and the ^{2}D master flats in a square region of half size 5 bins taken at the central X position of the detector, and Y position on the last seventh traced order, given from the input order table.

11.1.8 Interorder background subtraction

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

- 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 are computed in a subwindow of height (2 * **backsub.radiusy**/biny_size + 1). The window width is given by the distance between the sample positions. Spline interpolation of the sample points is performed to produce a background image, which is subsequently filtered using an average boxcar filter of size **backsub.smoothx** by **backsub.smoothy** pixels.

  The spline degree is set via **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 uves_cal_orderpos because the order locations, required for the spline method, are not known at this initial stage.

  The input image is sampled on a regular grid of **backsubgrid** sampling points each equi-spaced every (image_width **/ backsubgrid**) and (image_height **/ backsubgrid**) pixels along the X and Y directions. Using a window of height (2***backsubradiusy**+1) and width (image_width **/ backsubgrid**), the background counts at each sample point are measured using the method specified by **mmethod**. A 2D polynomial surface of degree **backsubdegx** and **backsubdegy** is then fitted to the sample points and sigma-clipping of outliers is performed, where outliers are defined as absolute residuals that are greater than **backsubkappa** times the RMS of the fit.

11.1.9 Hough transform

The Hough transform is a robust algorithm for detecting multi-dimensional features in images and estimating their parameters [1].

The Hough transform is computed by sampling the input order flat image at a set of evenly spaced columns separated by **samplewidth** pixels. Each echelle order maps to a peak in the Hough image. Peaks are detected in the Hough image, and care is taken to avoid repeat peak detections. Accurate peak locations are calculated as the centroid of the area around the local maximum in the Hough image.

The function detects orders until the intensity of the next candidate drops to below a fraction **pthres** of the dimmest line.

An important parameter in order for the robust peak detection to work is the (approximate) interorder spacing.

This parameter 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 varies mildly as a function of the order number.

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

11.1.10 Order tracing

The parameter **use_guess_tab** has the following acceptable values:

- 0: Ignore the guess order table.
- 1: Use the guess order table to set the lower/upper Y values where orders are searched for.
- 2: Use the guess order table to try and fully match the input order guess table.

The order tracing is performed as follows:

- Firstly, each order is traced in both directions starting from the order center which is inferred from 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). This is done by fitting a 1D Gaussian in the y direction to the order line at x-positions separated by the parameter **tracestep** and taking the Gaussian peak as the position of the order line. The set of x and y coordinates so measured traces the order. The trace stops if the counts of the order line drop below the threshold defined by **minthresh** in an x-range determined by the parameter **maxgap**.

- Next, each order trace is fitted with a straight line, and the entire order is rejected if the RMS of the given order is larger than the maximum allowed RMS (**maxrms**) relative to the median RMS of all orders.
- Finally, a global polynomial of automatic degree is fitted to all order traces as a function of Y and order number, and **kappa**-sigma clipping is applied to remove outliers. Alternatively, the user can define polynomial degrees using the parameters (**defpol1, defpol2**).

11.1.11 Line search

Emission lines in the extracted arc lamp spectrum are searched for such that the detected number of lines lies in the range [**minlines**,**maxlines**] (both inclusive). This is achieved by adjusting the detection threshold until an appropriate number of lines is detected. Information relating to the set of detected lines is written to the columns "Y" (the relative order number), "X" (the line position in pix), "Xwidth" (the detected line sigma) and "Peak" (the peak counts) of the output line table(s), `LINE_TABLE_<CHIPID>`.

An emission line is detected if its peak counts are greater than the threshold plus the local background level, which is defined as the median of a window of width (2 * **range** + 1) centered on the line position. If flat fielding was performed, then the threshold is the line peak in units of sigma.

After an emission line is detected, the line position is calculated performing a 5-parameter Gaussian fit, including the continuum slope.

Finally, doublets (defined as emission lines with positions within 2.0 pixels) are removed from the set of detected lines.

11.1.12 Determination of the wavelength calibration

As a first step, the set of detected lines in the extracted arc lamp spectrum are cross-correlated with the guess line table, in order to determine a systematic x-shift of up to **shiftmax** pixels in either direction. The initial dispersion relation obtained in uves_cal_predict by fitting the relation

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

to the guess line table, which contains the relation from (x, m) = (pixel, order) to wavelengths (here f is a 2D polynomial), is then adjusted by the systematic x-shift. The resolution of the cross-correlation function is defined by **shiftstep**. The parameter **shifttoler** defines the tolerance in pixels for the line matching. The default polynomial **degree** is 4.

The full wavelength calibration starts from the adjusted first guess dispersion solution.

Line identifications with the line catalogue are made based on a match between the detected line predicted wavelength, λ_{com} and a catalogue wavelength, λ_{cat}. A successful line 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 neighbour (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: $\mathbf{tolerance} < \mathbf{ALPHA} * |\lambda\_\mathbf{cat} - \lambda\_\mathbf{nn}|$.

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

Next, a full wavelength solution is fitted using a polynomial model of degree **degree**. If **degree** is set to a negative value, then the polynomial degree is automatically adjusted based on the line residuals.

The above line identification procedure and wavelength solution fitting is iterated until no new line identifications can be made. After the first convergence of these iterations, all line identifications are cleared, and the whole fitting loop repeats starting with the last wavelength solution, but this time ignoring lines with residuals worse than **tolerance** and lines with residuals worse than **kappa** sigma.

11.1.13 Optimal extraction

The pipeline uses optimal extraction to achieve a higher signal-to-noise (S/N) in the extraction 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 [20] and Mukai [21]).

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 in the spatial direction (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 [21], 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. [14] The parameter **reduce.extract.profile** defines the kind of spatial profile to use. See also Figure 11.2.
- 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 [20]) assumes that the sky background has been already subtracted, and furthermore that the interpolated

[14] While resampling of data is often avoided because it introduces resampling noise [20], in this case the resampling noise is smoothed by 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".

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 generalise 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

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

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) 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

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

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

We also include a map of the flat field blemishes (due to dust, etc.) within the cosmic ray mask to discount these pixels throughout the extraction process. These blemishes are detected on the master flat field by comparison of the master flat field with a smoothed version of itself.

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 11.4). However, at higher S/N there is often a significant mismatch between the analytical profile and the empirical profile (see Figure 11.2) which is known to bias the extracted flux [20] and make the rejection of cosmic rays unreliable.

In order to have a fully automatic data reduction with guaranteed high 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

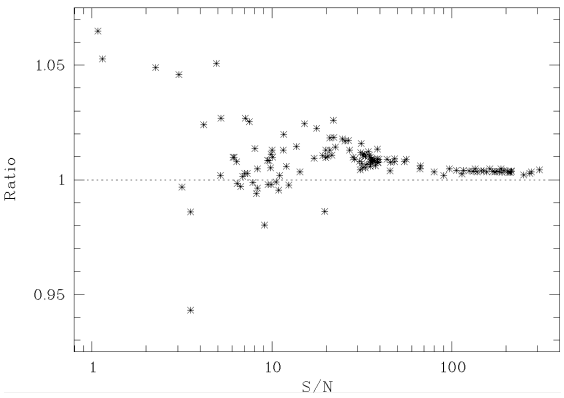


Figure 11.1: Ratio of the S/N obtained with the optimal sky subtraction and the S/N obtained with a separate, initial sky subtraction. Each point in the plot corresponds to the extraction of one echelle order in various data sets with different count levels.

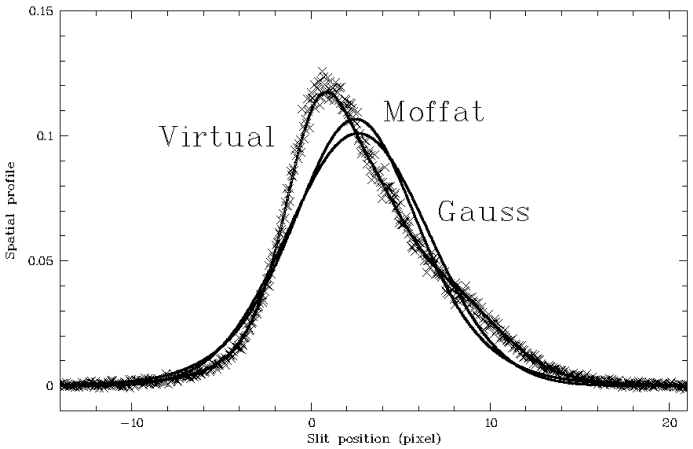


Figure 11.2: Empirical spatial profile (crosses) and three models (lines). At intermediate or high S/N the analytical methods usually cannot fit the object spatial profile accurately. The UVES instrumental PSF is known to be slightly asymmetric, but the high degree of asymmetry in the present data set, chosen here for illustration purposes, is due to the source being double.

than 10, then 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 may be repeated, but using the science frame.

Also, the degrees of all two-dimensional, low-order polynomials involved in the extraction process are determined at runtime by starting from (0, 0) degrees and increasing the degrees in steps of one or two as long as the fit 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.

For ways to change the position and/or size of the extraction see Sect. 11.2.8.

Assessing the extraction quality: The default values of the optimal extraction recipe parameters have been chosen 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 11.5 and 11.6). 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

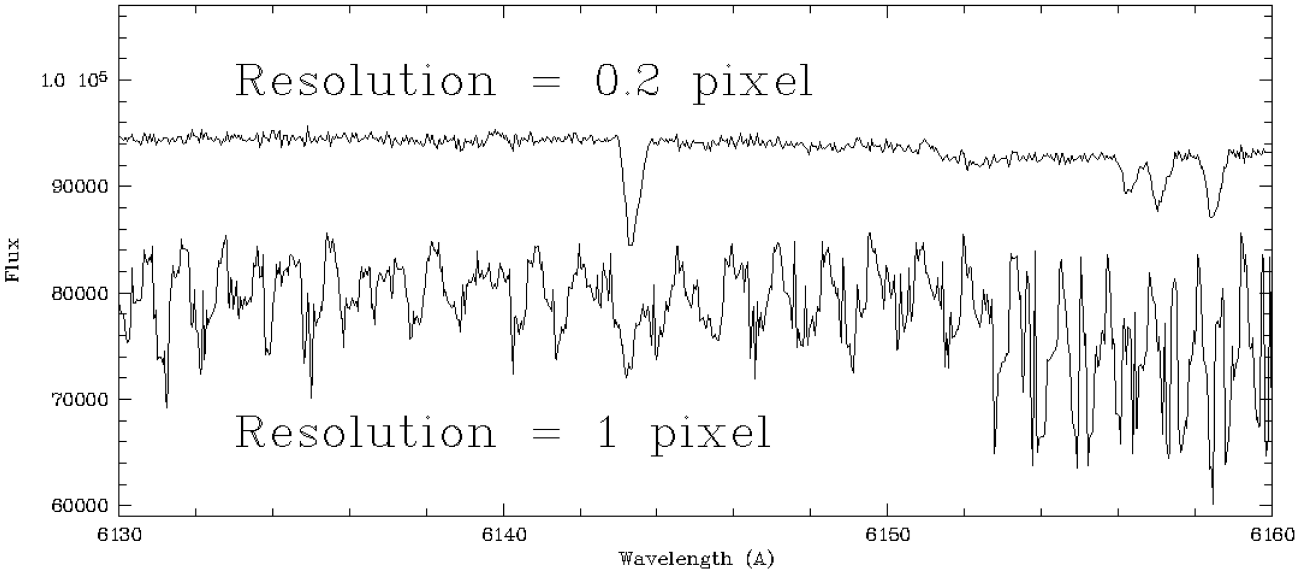


Figure 11.3: Zoom of a spectrum with S/N $\approx$ 290. Using a model profile with a resolution that is too low may result in a quasi-periodic pattern in the extracted spectrum.

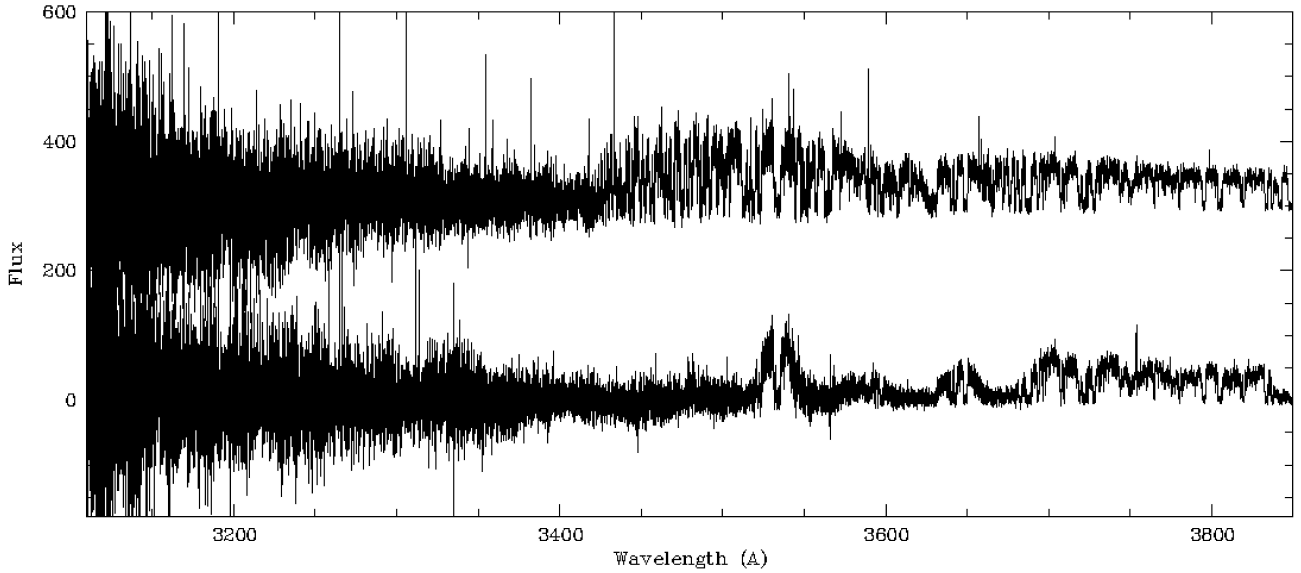


Figure 11.4: Comparison of an extracted low S/N QSO spectrum obtained using the analytical spatial profile algorithm (upper spectrum, shifted +300 units for clarity) with the same spectrum as obtained using the empirical spatial profile algorithm (lower spectrum). The latter presents regions of missing signal because the object tracing fails in some orders. The global model used in the current implementation facilitates the location and extraction of the object in every order in a robust manner.

was able to measure the spatial profile, exactly which cosmic rays were detected etc. and allows fine-tuning of the recipe parameters so that a close resemblance to the input frame is achieved.

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

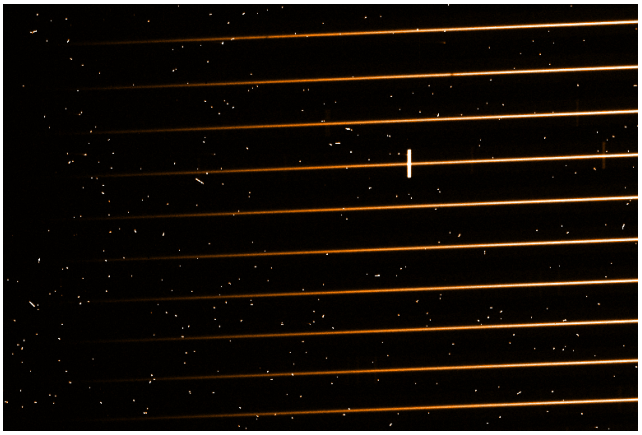


Figure 11.5: Input science image (detail).

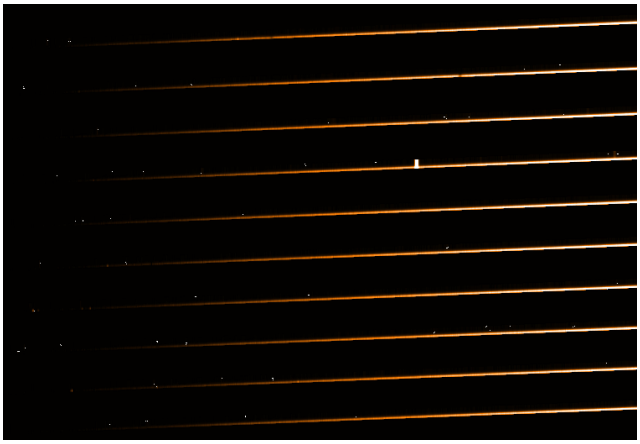


Figure 11.6: Science image reconstructed from the extracted spectra and cosmic ray map. For illustration purposes the spectrum was extracted using a short slit (1/4 of full slit) with a small positive offset (+1/8 of full slit). This reconstructed image shows the locations of the detected cosmic rays, 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.

11.1.14 Linear extraction

In linear extraction, for each order and wavelength the object pixel counts 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. For more details on specifying the extraction and sky windows see Sect. 11.2.8.

11.1.15 Average extraction

The average extraction is the same as the linear extraction, but the resulting flux is divided by the slit length in pixels. For more details on specifying the extraction and sky windows see Sect. 11.2.8.

11.1.16 Spatial extraction.

The so-called spatial (or 2D) extraction is a spectrum extraction algorithm which keeps the full spatial spectrum resolution. It is recommended for the extraction of object spectra whose cross-order FWHM is of the order of 1/3 or more of the observed slit length, or in the case that more than one object spectrum is observed along the slit length and the object spectra are blended with one another. This extraction method is activated by setting the parameter **reduce.extract.method** to "2d". If **reduce.ffmethod** is set to "pixel", then the object spectrum is pre-divided by the flat field.

Next 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", then the re-oriented object image is divided by the re-oriented flat field image. If **reduce.ffmethod** is set to "no", then 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/or **reduce.rebin.wavestep_redu** for REDU data) 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-flat-fielded spectra) or "optimal" (average weigthed by the inverse flux variances, appropriate for flat-fielded spectra) defined by the parameter **reduce.merge**.

11.1.17 Reduction of more than one 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, then the user can employ optimal or average extraction iteratively and set the slit offset and size to get the best extraction quality for each object. The extraction slit length and offset are defined by the parameters **reduce.slitlength** and **reduce.objoffset** respectively.

11.1.18 Reduction of observations acquired using the image slicer

The UVES spectrograph supports also an image slicer observing mode. In case of average or linear extraction the pipeline sets the extraction slit length set according to the slicer id (8 arcsecs in case of slicer #1 or #2 and 10 arcsecs in case of slicer #3) and performs no sky subtraction (if the image slicer is used there is not enough space on the detected spectrum to evaluate accurately the sky).

If the user reduces the data with default parameters an optimal extraction is performed with a fixed extraction slit length set as specified above. Optimal extraction would be extremly complex to perform as the cross order light profile of a spectrum acquired with the image slicer is unknown and difficult to model. To warn about this, the pipeline issues in case of optimal extraction of slicer data the following warning:

```
[WARNING] SciRed: Going to optimally extract an extended object (SCI_SLICER).
This may not work because the sky cannot be reliably determined
```

We advise the user to adopt the linear or the 2d method as extraction method for slicer data. The 2d method will provide a 2-dimensional rectified spectrum, from which the individual slicer spectra may be extracted with routines outside the pipeline.

11.1.19 Response computation

The observed STD star is first corrected for bias and inter-order background, flat fielded, optimally extracted and wavelength calibrated.

Then if the reference standard catalogue contains a spectrum with RA, DEC values within 60 arcsec from the corresponding values of the observed reference standard star, the instrument response is computed as described below.

Initially the flux table corresponding to the observed standard star is extracted from an input catalogue. We correct the wavelength scale of the reference spectrum (which is a stellar model spectrum) to the same radial velocity as the observed spectrum and then interpolate the reference spectrum to the same steps as the observed spectrum.

The response is computed by dividing this model spectrum of the standard star (in erg $\mathrm{cm}^{-2}\mathrm{s}^{-1}\mathrm{\AA}^{-1}$) by the 1D extracted observed spectrum of the standard star corrected for gain, exposure time, atmospheric extinction (the atmospheric extinction table is interpolated to get the same binning). To reduce the noise of the resulting ratio spectrum we apply a median filter of eleven pixels half width. Then we apply a cubic-spline fit to the points defined in RESP_FIT_POINTS_CATALOG to get the final response.

The response is obtained with the following equation:

$$\mathrm{Response[erg/e^{-}/cm^{2}]} = \frac{\mathrm{STD_{fluxtable}[erg/s/cm^{2}/\mathrm{\AA}^{-1}]} \times \mathrm{exptime[s]} \times \mathrm{gain[ADU/e-]}}{\mathrm{STD_{observed}[ADU/pix[\mathrm{\AA}]]} \times 10^{(0.4 \cdot \mathrm{airmass} \cdot \mathrm{ext})}} \tag{6}$$

Where we have taken into account of the size of a pixel. The response is derived from the order merged flux standard spectrum and applied on the merged science spectrum.

11.1.20 Flux calibration

If the user provides the instrument response and the atmospheric exctiontion tables in input of a science recipe, the merged 2D and 1D spectra are then flux calibrated. This operation is performed by first dividing the observed spectra by exposure time and the detector gain and correction for atmospheric extinction, and then multiplying those by the instrument response.

$$\mathrm{I[erg/s/cm^{2}/\mathrm{\AA}]} = \frac{\mathrm{I[ADU/pixel[\mathrm{\AA}]]} \times \mathrm{Response[erg/e^{-}/cm^{2}]} \times 10^{(0.4 \cdot \mathrm{airmass} \cdot \mathrm{ext})}}{\mathrm{gain[ADU/e-]} \times \mathrm{Exptime[s]} \times \mathrm{bin\_size}} \tag{7}$$

Where we evidence the fact we correct for the size in pixels of the integration bin.

11.1.21 Efficiency computation

The recipe uves_cal_response computes the instrument response curve aswell 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 resolution by 3rd order spline Hermite interpolation.

After computation of the effective counts per erg ($ctsperg$), 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 11.15).

11.1.22 Spectrum flux calibration

To flux calibrate a spectrum the recipe first corrects the spectrum for exposure time, gain, binning, atmospheric extinction, and airmass and then multiplies it with the input response (if available a master response). Beforehand the response curve is rebinned to the observed spectrum via Hermite polynomial interpolation.

11.2 Recipes

In the following sections we will describe the recipes in the UVES pipeline. Common recipe parameters are: **debug**, to activate the debug mode which saves intermediate results to disk, **process_chip**, to allow RED arm data reduction only on one of the two chips of the red detector mosaic, **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 corresponding algorithm description.

11.2.1 uves_cal_mbias

This recipe generates a master bias frame from a set of raw bias frames. Each bias frame is corrected by the sigma-clipped mean of all the pixels and then they are combined by median combining, if **stack_method=** "median", or by kappa-sigma-clipped mean combination (see Sec. 11.1.2), if **stack_method=**"mean". Finally the mean bias level is added back into the master bias.

The detector read-out noise (RON) is measured in a central box 100 $\times$ 100 pixels in size, both on the raw and the master bias frame. The RON should decrease as $\sqrt{2N/\pi}$ for median combination, where N is the number of combined frames, and the factor $2/\pi$ is a result of the use of the median to combine the frames.

If the parameter **clean_traps** is set to TRUE, then detector traps are linearly interpolated using the pixel values in the adjacent columns. The detector trap locations are known and hard coded in a pipeline C function. The known positions are reported in the following table:

DETECTOR	CHIPID	BINX	BINY	LLX	LLY	URX	URY
NEW	REDL	1	1	1	4	2088	4
OLD	REDL	1	1	1	4	2088	4
NEW	REDL	1	1	1	63	2282	63
OLD	REDL	1	1	1	63	2282	63
NEW	REDL	1	1	1	108	1778	108
OLD	REDL	1	1	1	108	1778	108
NEW	REDL	1	1	1	176	2443	176
OLD	REDL	1	1	1	176	2443	176
NEW	REDL	1	1	1	196	2021	196
OLD	REDL	1	1	1	196	2021	196
NEW	REDL	1	1	1	285	1974	285
OLD	REDL	1	1	1	285	1974	285
NEW	REDL	1	1	1	352	1942	352
OLD	REDL	1	1	1	352	1942	352
NEW	REDL	1	2	1	4	1045	4
OLD	REDL	1	2	1	4	1044	4
NEW	REDL	1	2	1	63	1141	63
OLD	REDL	1	2	1	63	1141	63
NEW	REDL	1	2	1	108	894	108
OLD	REDL	1	2	1	108	894	108
NEW	REDL	1	2	1	176	1222	176
OLD	REDL	1	2	1	176	1222	176
NEW	REDL	1	2	1	196	1011	196
OLD	REDL	1	2	1	196	1011	196
NEW	REDL	1	2	1	285	988	285
OLD	REDL	1	2	1	285	988	285
NEW	REDL	1	2	1	352	971	352
OLD	REDL	1	2	1	352	971	352
NEW	REDL	2	2	1	4	1045	4
OLD	REDL	2	2	1	3	1044	3
NEW	REDL	2	2	1	14	1255	14
NEW	REDL	2	2	1	33	1141	33
OLD	REDL	2	2	1	33	1141	33
OLD	REDL	2	2	1	55	894	56
NEW	REDL	2	2	1	89	1222	89
OLD	REDL	2	2	1	89	1222	90
NEW	REDL	2	2	1	99	1011	100
OLD	REDL	2	2	1	99	1011	100
NEW	REDL	2	2	1	144	988	144
OLD	REDL	2	2	1	144	988	145
NEW	REDL	2	2	1	177	971	178
OLD	REDL	2	2	1	177	971	178
NEW	REDL	2	3	1	3	696	3
OLD	REDL	2	3	1	3	696	3
NEW	REDL	2	3	1	14	836	15
NEW	REDL	2	3	1	33	761	33
OLD	REDL	2	3	1	33	761	33
NEW	REDL	2	3	1	55	596	56
OLD	REDL	2	3	1	55	592	56
NEW	REDL	2	3	1	89	814	90
OLD	REDL	2	3	1	89	814	90
NEW	REDL	2	3	1	97	805	90
NEW	REDL	2	3	1	99	674	100
OLD	REDL	2	3	1	99	674	100
NEW	REDL	2	3	1	144	658	144
OLD	REDL	2	3	1	144	658	144
NEW	REDL	2	3	1	177	971	178
NEW	REDL	2	3	1	156	784	156
NEW	REDL	2	3	1	168	759	168
NEW	REDL	2	3	1	177	647	178
OLD	REDL	2	3	1	177	647	178
NEW	REDL	2	3	1	203	826	203

NEW	REDL	2	3	1	263	714	263
NEW	REDU	1	1	1	845	1268	845
OLD	REDU	1	1	1	2030	1268	2033
OLD	REDU	1	1	1269	2033	4096	2033
OLD	REDU	1	1	1201	491	3271	492
NEW	REDU	1	2	1	1396	845	1396
OLD	REDU	1	2	1	2030	634	2033
OLD	REDU	1	2	635	2033	2048	2033
OLD	REDU	1	2	600	491	1635	492
NEW	REDU	2	2	1	422	1526	422
OLD	REDU	2	2	1	1013	634	1016
OLD	REDU	2	2	635	1015	2048	1016
OLD	REDU	2	2	600	244	1635	245
NEW	REDU	2	3	1	61	287	62
NEW	REDU	2	3	1	422	1051	422
NEW	REDU	2	3	400	872	1265	872
OLD	REDU	2	3	1	1013	423	1016
OLD	REDU	2	3	1	1015	1365	1016
OLD	REDU	2	3	400	244	1090	245

The parameters QC.OUT1.STRUTCTX and QC.OUT1.STRUCTY provide a way to monitor more accurately the stability of a frame. Their values are obtained by collapsing the frame along a given axis and calculating the RMS. A sudden change in these parameters may indicate a local change of the master frame. The collapsed frame may be displayed in a 2D plot to spot local frame inhomogeneity or changes with time.

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 shown in Figure 11.7.

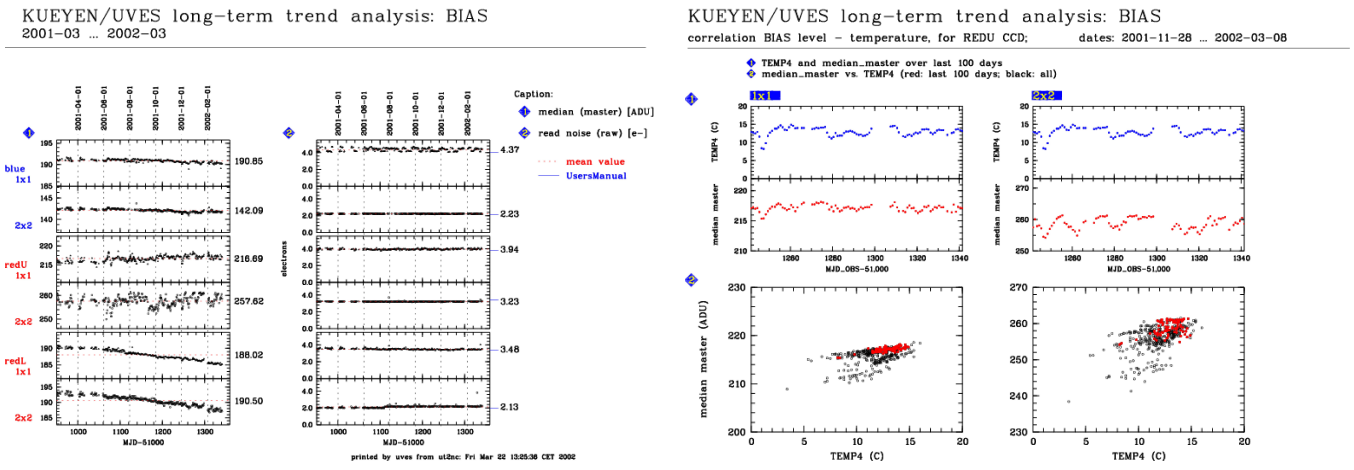


Figure 11.7: Trending plot of the median value for a master bias and the noise computed on a raw bias for different instrument settings (left panel). The REDU CCD shows pronounced fluctuations around a stable mean value which are at least partly correlated with temperature. In the trending plot (for 1x1 and 2x2 binnings) shown in the right panel, the first row shows the temperature (TEMP4), and the second row shows 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.

11.2.2 uves_cal_mdark

This recipe creates a master dark frame from a set of darks (darks are not frequently taken for UVES).

The master dark frame is created by subtracting the master bias frame from each input raw dark frame. Then the recipe combines the bias corrected dark frames by employing the median, if **stack_method**="median", or by employing a kappa-sigma-clipped mean (see Sec. 11.1.2), if **stack_method**="mean".

To allow the detection of parasitic light contamination which typically manifests itself as localised patterns of light on the dark frames, this recipe computes a set of quality control statistics over a number of independent regions distributed over the chip. In this way it is possible to detect any parasitic light contamination under the assumption that one or more regions will show abnormal behaviour while others will remain unaffected. The statistics are computed for each input raw dark frame (**qc_dark.reg.when** = 1) or for the master dark frame (**qc_dark.reg.when** = 0). The relevant dark frame is divided into **qc_dark.reg.num_x** by **qc_dark.reg.num_y** regions each of size **qc_dark.reg.box_sx** by **qc_dark.reg.box_sy** pixels. The user may leave a space between one region and the next one by setting border parameters **qc_dark.reg.border_x** and **qc_dark.reg.border_y** for the x and y coordinate directions respectively. In each region, the maximum, minimum, mean, median and RMS values of the set of pixels are calculated. Also, for the set of regions, the maximum, minimum, mean, median and RMS values of the set of statistics are calculated.

11.2.3 uves_cal_predict

This recipe expects as input a format-check frame obtained by illuminating a narrow slit with a ThAr lamp, and a reference table of ThAr lines. Optional additional input is a reference format-check frame, to compute spectral format shifts relative to a reference position. The recipe is used to generate guess order and line tables, which subsequently allow automatization of the order and wavelength calibration steps. The guess line table is also used to monitor the spectral format against predictions from the physical model and aid in the detection of major changes. Quantities such as the shift of the current frame relative to a reference frame (11.1.6) or the number of detected lines are used for such purposes. For example, by trending the spectral format shifts as a function of the temperature, it has been possible to quantify and automatically compensate (with the instrument software) for thermal drifts in the X and Y directions (see Figure 11.8).

As shown in Figures 11.9 and 11.10, the recipe uves_cal_predict allows one to monitor the spectral format against the model prediction and identify if a major shift has occurred (as in the case shown in the central panels of Figure 11.10), which may be the consequence of a major earthquake event. Such a change may trigger an instrument re-alignment (to the reference position indicated by the reference format-check frames).

The recipe performs the following data reduction steps:

- The input format-check frame is correctly re-orientated (rotated, split and flipped if needed as in the RED arm case).
- Relevant ambiental and instrument setting information along with the reference line table are read in to be used to compute the predicted line positions.
- Line positions are predicted from the UVES physical model (see 11.1.3) and matched to the actual frame line positions (see 11.1.4).

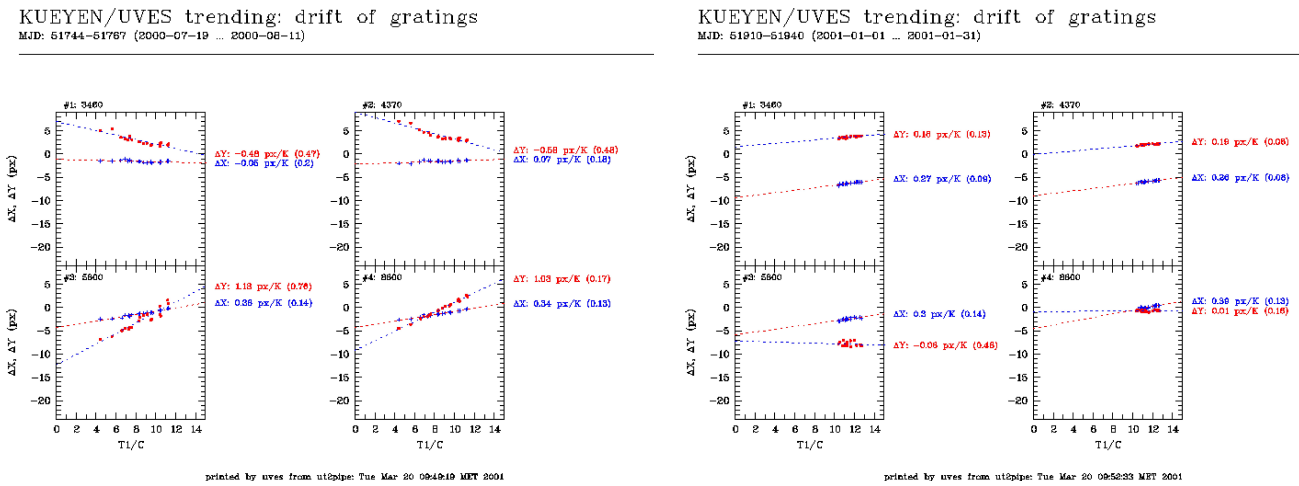


Figure 11.8: Thermal drift of the UVES gratings before (left) and after (right) the compensation of the Y shifts introduced on Nov 2000.

- Guess order and line tables are created (see 11.1.5).
- If a reference format check frame is available, then a stability check is performed, by computing guess line and order tables for the current and reference format check frames and by computing the corresponding shifts by performing a table cross-correlation (see 11.1.6).
- The computed average and median X and Y shifts, together with the ambient temperature, pressure and instrument central wavelength setting are logged as QC parameters.

The line guess table may be used to generate a set of very useful quality control plots (see Figure 11.9).

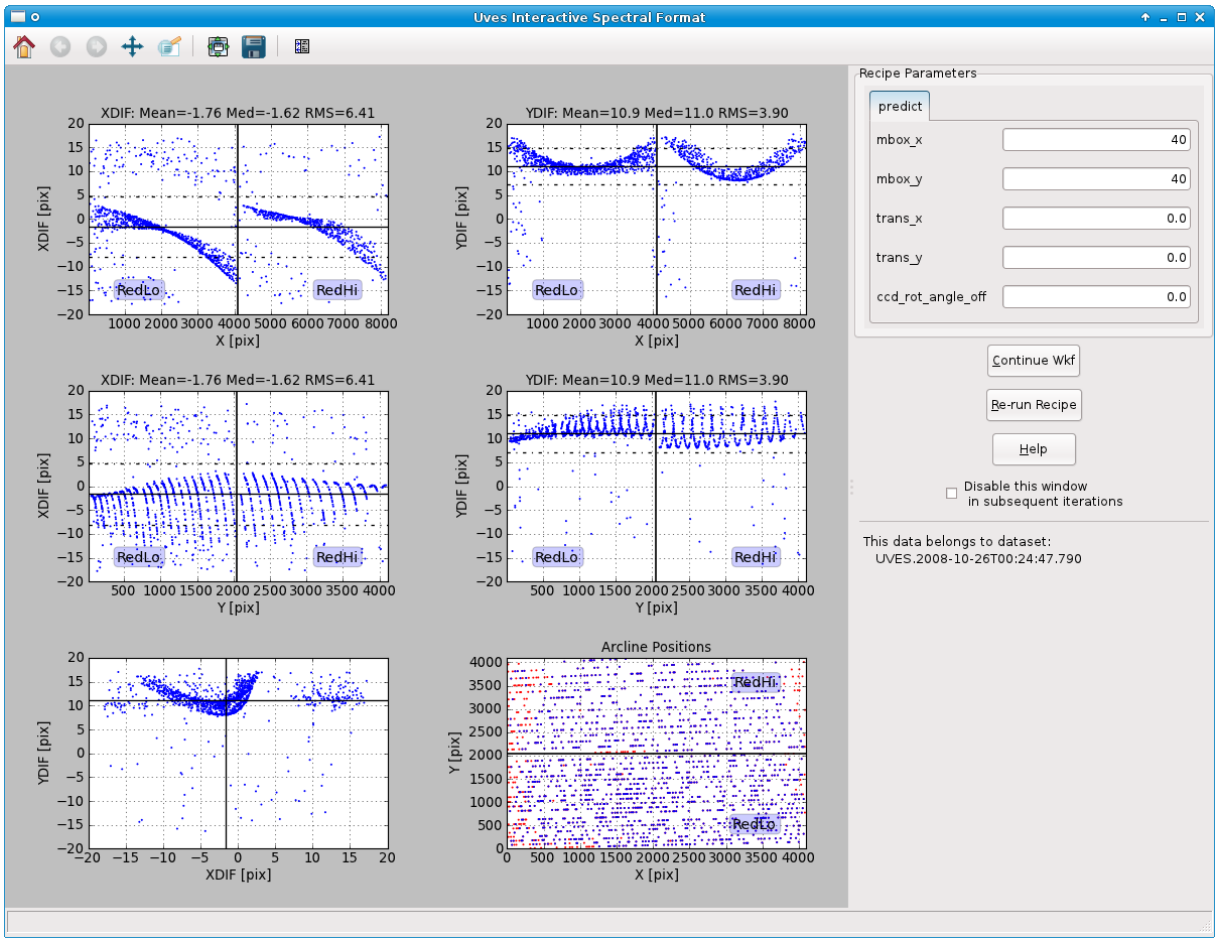


Figure 11.9: Physical model plots as appearing in the UVES-Reflex workflow. Top-left: difference between predicted and measured X positions as a function of predicted X. Top-right: difference between predicted and measured Y positions as a function of predicted X. Center-left: difference between predicted and measured X positions as a function of predicted Y. Center-right: difference between predicted and measured Y positions as a function of predicted Y. Bottom-left: difference between predicted and measured Y positions as a function the difference between predicted and measured X positions. Bottom-right: detected lines positions. A well defined distribution with mean ordinate zero is an indication of good matching between the model predictions and the line positions in a format check frame.

As described in Ballester et al., "The UVES Data Reduction Pipeline", ESO Messenger No. 101, this recipe is successful if the plots in Figure 11.9 (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 with mean ordinate value equal to zero (see Figure 11.10). Consequently, in the case of a scattered distribution of points, the physical model predictions are not appropriate for the actual format-check frame under consideration. This may occur in the case of an instrument set-up misalignment (for example, induced by a strong Earthquake) which results in line shifts along Y that are greater than $\sim$10 pixels. Usually the physical model is robust enough for the recipe to find a good solution for shifts of less then $\pm$10 pixels in X and Y.

We now proceed to describe how to recover a good solution during interactive data reduction for this recipe. The

following procedure should be attempted whenever one receives data which give residual plots with a complete scatter of points (for example, see Figure 11.10(b)&(d)), or if the instrument spectral format has drifted more than ∼7-10 pixels, or if the user has required data with a non-standard setting.

Note that standard settings are considered to be 1x1 and 2x2 binning, and central wavelengths of 520, 564, 580, 600, 760 and 860 nm in FIBER mode. After the red arm CDD detector upgrade the pipeline automatically sets **ccd_rot_angle_off** to compensate for a small tilt of the upper chip with respect to the previous configuration. The pipeline will also attempt to automatically adjust **trans_x** and **trans_y** for the following additional non-standard central wavelength settings: 500, 540, 560, 580, 620, 640, 660, 700, 720, 740, 780, 800, 880 and 900 nm; and for the non-standard binnings of 1x2 and 2x3 for the central wavelength settings 520, 564, 580, 600, 760 and 860 nm.

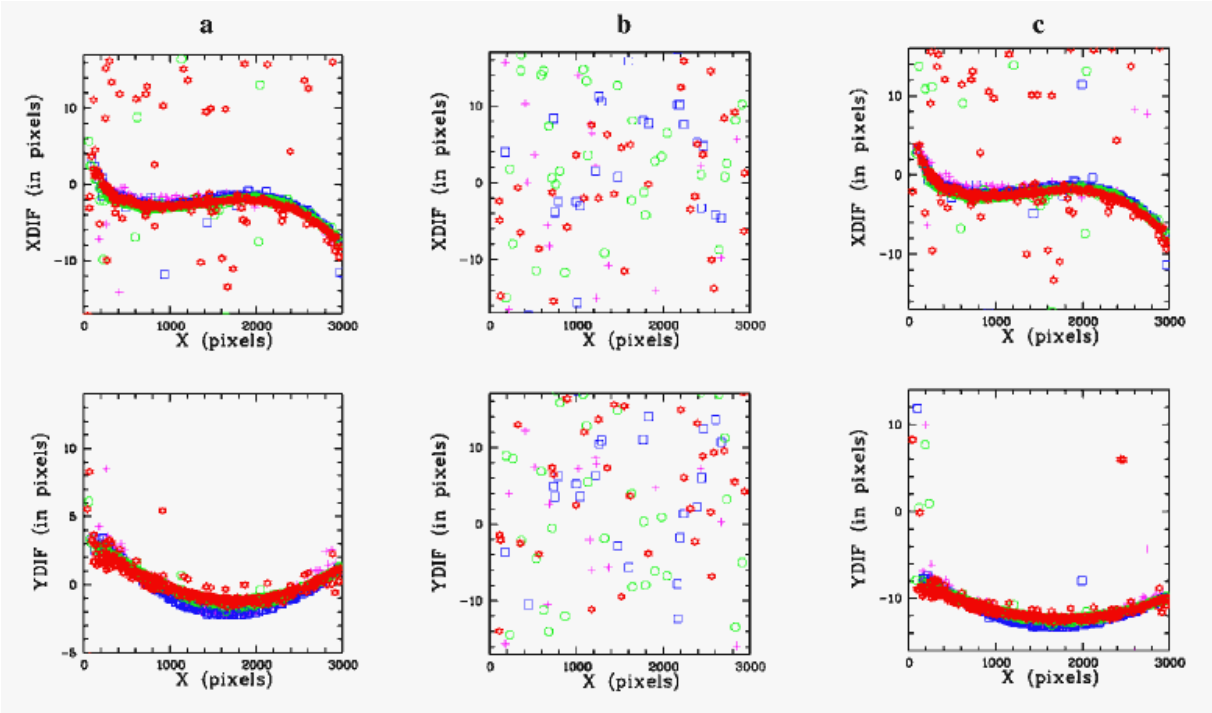


Figure 11.10: *The May 12, 2000 earthquake event as detected from the physical model control plots. The normal result obtained after successful line matching (a) produces a well concentrated distribution with mean ordinate zero. The earthquake event causes the lines matching recipe to fail (b). Adjusting the model by -10 pixels (along the cross-order direction) again matches the instrument configuration (c)..*

The parameters to be modified to improve the UVES physical model accuracy are **mbox_x, mbox_y, trans_x, trans_y** and **ccd_rot_angle_off**. Other parameters affecting the accuracy, but that are not expected to be changed, are **ech_angle_off** and **cd_angle_off**.

In order to manually optimize the physical model guess solution for a given format-check frame one should carry out the following steps:

1. Use a reference format-check frame, which is a frame that results in residual plots with a very good aggregation of points. This frame should have the same instrumental setting as the current format-check

frame and the value of TAG (and ESO.PRO.CATG) must be set to MASTER_FORM_x (x = BLUE or REDL, REDU). Reference format-check frames can be found in the directory fluves/calib of the pipeline release tar file. Two sets of reference format-check frames are available: one defined at the start of UVES operations, and one, including only RED arm data, defined at the time of the UVES RED arm CCD upgrade. The user must use coherent data sets: the reference format-check frames from the year 2000 in the case of pre-detector upgrade format-check data recovery, and the other for newer data.

2. Obtain a first approximation of the X and Y shifts **trans_x** and **trans_y** by manually comparing the reference and current format-check frames.

3. Run the recipe uves_cal_predict on a set of frames containing the current format-check frame, the reference format-check frame(s) (one for each detector CHIP ID: BLUE or REDL and REDU), and the reference arc line catalogue, setting the values of **trans_x** and **trans_y** to those values determined in the previous step. Note that in the first iteration the user may need to set the recipe input parameters **mbox_x** and **mbox_y** (the match window size) values to greater than the default values (for example, 60 or 80 instead of 40).

4. Repeat the previous step by adding the values of QC.MODEL.DIFFXMED and QC.MODEL.DIFFYMED reported in the paf files `lineguesstable_<chipid>-0.paf` (respecting their sign) to the input values of **trans_x** and **trans_y**. Stop the iterative process when the values of QC.MODEL.DIFFXMED and QC.MODEL.DIFFYMED are close to zero (and certainly less than 1 pixel).

5. Try to reduce the **mbox_x** and **mbox_y** values gradually to the default value (40), and then repeat the previous step.

11.2.4 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 order detection, a guess order table. The product is an order table.

The recipe performs the following data reduction steps:

- A median boxcar filter of size $(2 \times \textbf{radx} + 1)$ by $(2 \times \textbf{rady} + 1)$ pixels is applied to the input flat-field frame.

- The interorder background is subtracted using the polynomial method (see section 11.1.8).

- If no guess order table is provided, then the orders are detected by means of a Hough transform (see 11.1.9).

- The orders are then traced (see 11.1.10), and the traces 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%.

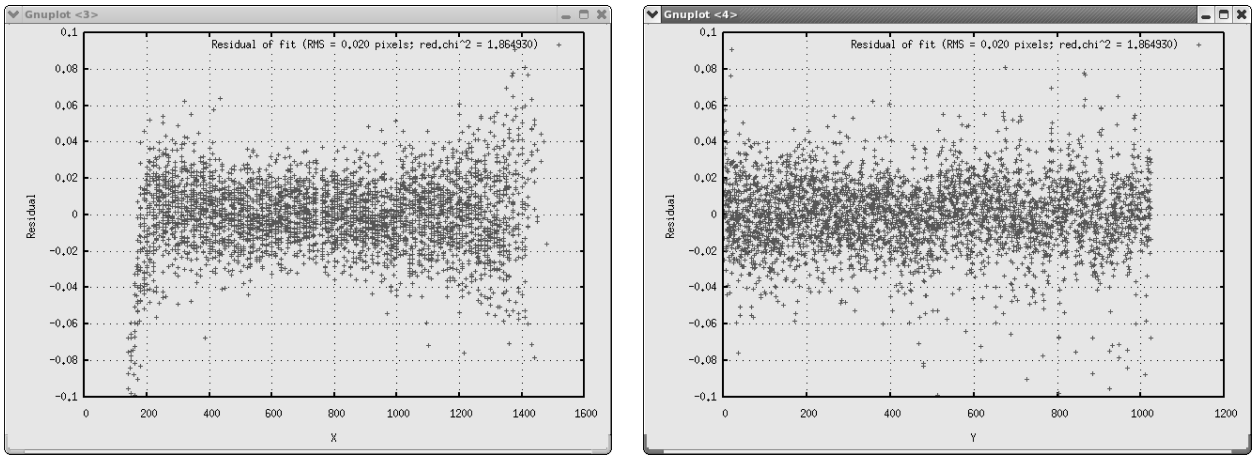


Figure 11.11: Order definition residual plots as a function of X (left) and Y (right).

11.2.5 uves_cal_mflat

This recipe generates a master flat frame from a set of flats by performing the following data reduction steps:

1. Corrects the flat frames for bias and dark.
2. Determines the exposure level of each flat frame by determining for each order the mean exposure level over ten sampling windows uniformly distributed along the order and then taking the mean of these values over the orders. Each flat frame is then normalised by its exposure level, and the normalised flat frames are combined into an initial master flat frame by median combining.
3. Divides the initial master flat frame from (2) into each calibrated flat frame from (1). Now each and every pixel in the resulting flat-fielded flat frame gives an estimate of the exposure level of the flat frame, regardless of the order it belongs to. Then, to estimate the exposure level of each flat frame, the recipe calculates the median of all pixels in all orders on the flat-fielded flat frame.
4. Normalises each calibrated flat frame from (1) by the exposure levels derived in (3), and combines the flats using the median to obtain the final master flat frame.
5. Finally the master flat frame is also background subtracted using the spline method (see section 11.1.8).

11.2.6 uves_cal_wavecal

This recipe performs the wavelength calibration using previously determined solutions for the guess line table and the order table. The slit is divided into **nwindows** (default three) subslits of length **length** (default equal length), and a wavelength calibration solution is determined for each of the subslits (at a given **offset**, defaulted to 0).

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

- If a master bias is provided, then it is subtracted from the arc lamp frame.
- If a master flat is provided, then the bias corrected arc lamp frame is divided by the master 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, then the resulting 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 11.1.15, 10.8.5).
- Emission lines are detected in the extracted spectrum (See 11.1.11).
- 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 guess line table, and then apply this shift to the guess dispersion relation (see 11.1.12).
- Iterative line identification and update of the dispersion relation, until no new line identifications can be made (see 11.1.12).
- The extracted spectrum is then wavelength calibrated using a bin size of **rebin.wavestep** (or of **rebin.wavestep_redu** for REDU data) (defaulted to 2/3 of the average pixel size of the corresponding detector chip) and the orders are merged. The user also has the option of applying the flux scaling factor $\frac{dx}{d\lambda}$ (pixel per wavelength) during the rebinning process in order to convert the observed flux (in pixel space) to a flux per wavelength (in wavelength space). The spectral resolution is computed by determining the line width for each of the identified lines in the merged spectrum.

For quality control purposes, it is important to check the plot of wavelength residuals for the successfully identified lines listed in the reference line table (see Figure 11.12).

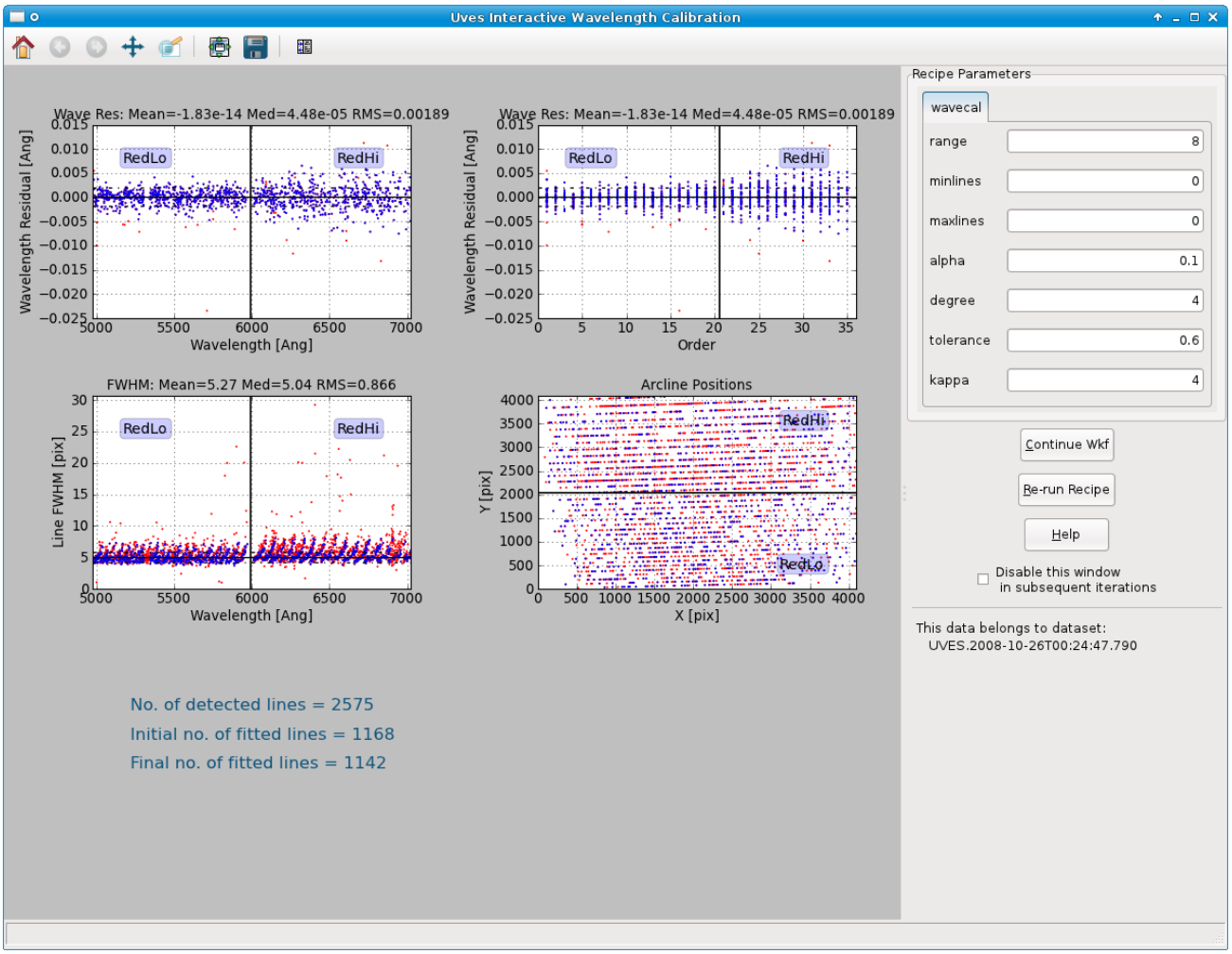
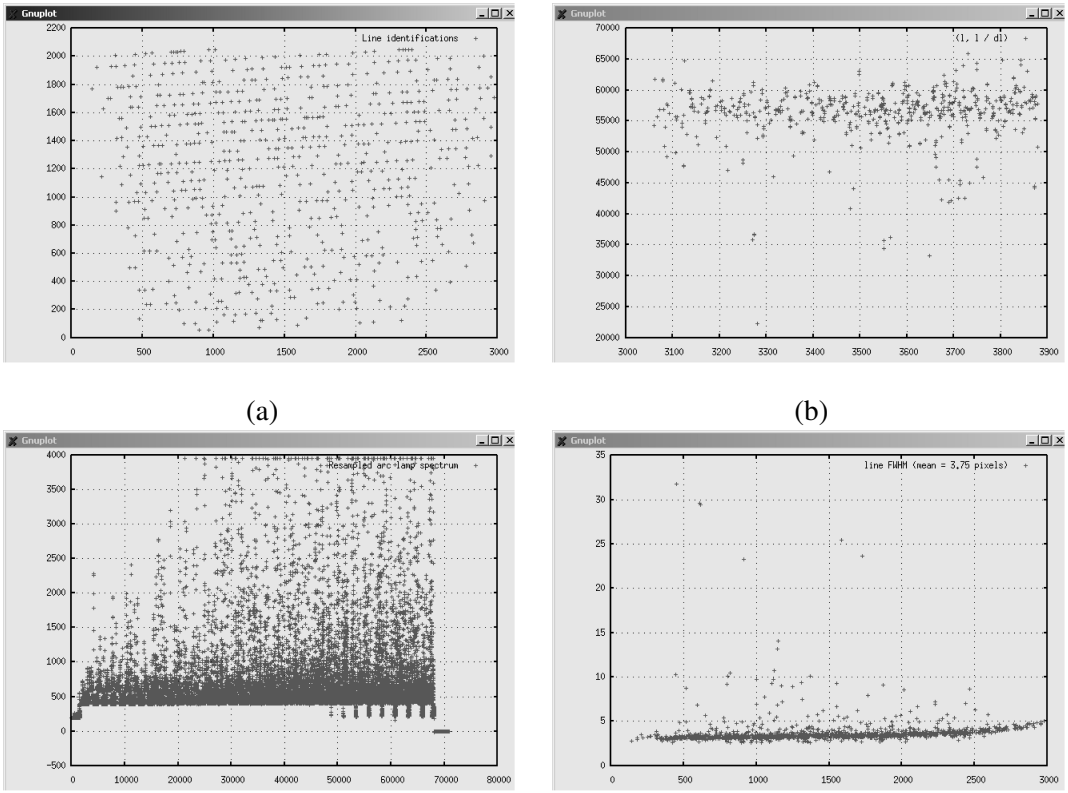


Figure 11.12: Reflex interactive wavelength calibration window results: upper-left: wavelength residuals [Å] as a function of wavelength [Å]. upper-right: wavelength residuals [Å] as a function of order number. bottom-left: wavelength FWHMs [Å] as a function of wavelength [Å]. bottom-right: detected (red) and fitted (blue) arcline positions [pix] on the detector.

Other plots that are useful to inspect are the distribution of identified lines across the detector, the resampled ThAr spectrum, the resolving power as a function of wavelength, and the line FWHM as a function of wavelength (see Fig. 11.13).

The wavelength calibration 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 11.14) the wavelength solutions 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 10.8.5) the line table contains three identical solutions, and the linear interpolation has no effect.



(a) (b)

(c) (d)

Figure 11.13: (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.

11.2.7 uves_cal_response

This recipe computes the instrument response function and the telescope+instrument+detector efficiency via the following steps:

- The standard star spectrum is optimally extracted (see 11.1.13).
- The standard star reference catalogue FLUX_STD_TABLE is searched for the relevant standard star entry by performing a match on the celestial coordinates of the standard star observation.
- If a matching entry was found in the standard star reference catalogue, then the response of the instrument is determined by aligning the reference spectrum to the observed one in wavelength and sampling and then calculating the ratio of the relevant spectrum from the reference catalogue to the standard star spectrum (see Section 11.1.19 and Figure 11.15 panels (a) and (b)).
- Finally, the efficiency of the telescope+instrument+detector is calculated (see Figure 11.15 panel (c) and Section 11.1.21).

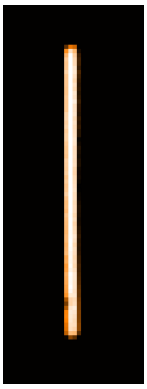
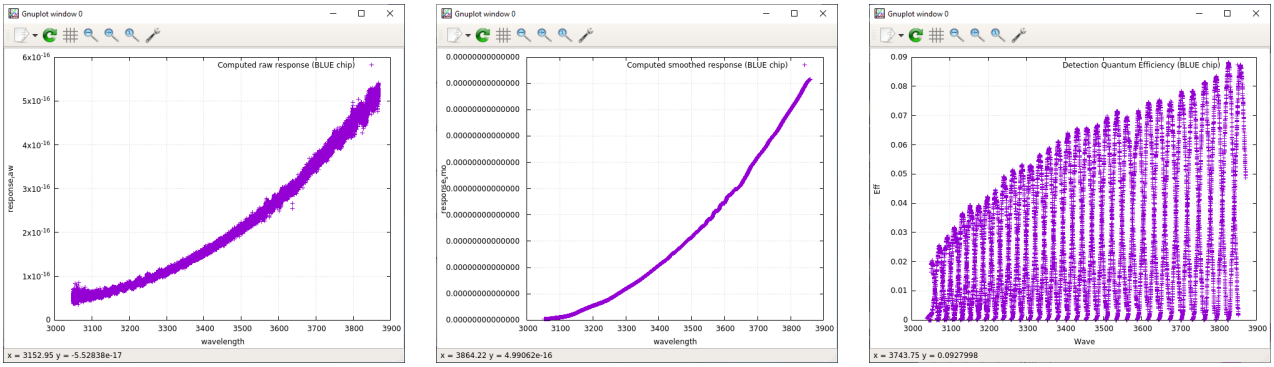


Figure 11.14: The cross-dispersion direction is not perfectly aligned with the detector columns. The centroid position of the ThAr emission line shown here (from the lower red chip, central wavelength = 860 nm) changes by up to ~ 0.5 pixels as a function of slit position. Since the wavelength calibration 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) (b) (c)

Figure 11.15: The UVES pipeline also produces instrument response curves (raw (a) and smoothed (b)) and plots of the instrument efficiency (c). The latter may be used to monitor the instrument health and, from time to time, upgrade the UVES Exposure Time Calculator predictions.

11.2.8 uves_obs_scired

This recipe reduces a science frame `SCIENCE_<ARM>` by using a combination of the following steps:

- If **clean_traps** is set to TRUE, then detector traps are linearly interpolated using the pixel values in the adjacent columns (see the table in Section 11.2.1 that specifies the location of known detector traps).
- The master bias is subtracted.
- The master dark (if provided) is subtracted.
- The interorder background is estimated using the spline method as described in 11.1.8, and then subtracted.
- If **reduce.ffmethod** is set to “pixel”, then the corrected science frame is divided by the master flat field.

- Then the object is extracted using either optimal extraction (11.1.13), linear extraction (11.1.14), average extraction (11.1.15) or the 2D method (11.1.16). If the spectrum is optimally extracted, cosmic rays are detected and rejected, and the sky is fitted and subtracted during the extraction. If linear/average extraction is used, then the sky is extracted in windows above and below the object and subsequently subtracted.
- If **reduce.ffmethod** is set to "extract", then the master flat field is extracted using the object weights obtained in the previous step, and the object spectrum is divided by the extracted master flat field spectrum.
- The spectrum is resampled to equidistant wavelength steps with a size defined by the parameter **reduce.rebin.wavestep** (for BLUE and REDL), and the parameter **reduce.rebin.wavestep_redu** (for REDU), and optionally (depending on **reduce.rebin.scale**) multiplied by $\frac{d\lambda}{dx}$ to ensure flux conservation.
- In the case that the extraction is average or linear extraction then the sky is subtracted if **reduce.skysub** is set to TRUE.
- By default (**reduce.slitlength** = -1) the full slit length for the extraction is obtained from the header keyword `HIERARCH ESO INS SLIT2 LEN` (blue arm) or `HIERARCH ESO INS SLIT3 LEN` (red arm) and the full slit is centered on the center of the order. The recipe parameters **reduce.slitlength**, **reduce.objoffset**, and **reduce.objslit** allow the user to fine-tune the extraction. Their usage depends on the extraction mode in use, as described below:

  Warning If the value specified for **reduce.slitlength** exceeds the value derived from the header keywords inter-order regions may be included in the extraction.

  In average/linear extraction:

  reduce.slitlength specifies the full slit length within which one object window and up to two sky windows will be defined

  reduce.objoffset [ignored if **reduce.objslit** <0 (default -1)] allows the user to offset the object window away from the order center

  reduce.objslit if >0 **reduce.objslit** specifies the size of the object window, which can then be offset from the order center via **reduce.objoffset**; if ≤ 0 (default), the object window size will be 1/2 of the full slit, and its position will be defined by searching the object's spectrum within the slit

  In optimal extraction:

  reduce.slitlength specifies the full slit length within which object and sky spectra will be extracted simultaneously

  reduce.objoff allows the user to offset the full slit as specified by **reduce.slitlength** away from the order center

  reduce.objslit completely ignored

  Key differences between the three modes are

  1. In average/linear extraction the full slit is always centered on the order center, but in optimal extraction it can be offset away from the order center by the **reduce.objoffset** parameter
  2. Optimal extraction always uses **reduce.objoffset** but completely ignores **reduce.objslit**, whereas average/linear extraction uses **reduce.objoffset** only if **reduce.objslit** is >0

3. In average/linear extraction, once a portion of the full slit has been reserved by the object window as described above, sky windows of size 0.5·(**slitlength**−**objslit**)±**objoffset** will be carved out of the remaining full slit (if any). **slitlength**, **objslit**, and **objoffset** denote the actual values of the length of the full slit, the size of the extraction window, and its offset from the slit center. This definition of sky windows yields the following possible results:

   2 sky slits both sky slits were non-negative, execution continues

   1 sky slit one of the sky slits was negative and discarded, execution continues using the other; this can occur if
   objslit $>$ **slitlength** or
   objoffset $>$ 0.5·(**slitlength** - **objslit**) or
   objoffset $<$ −0.5·(**slitlength** - **objslit**)

   0 sky slits both sky slit sizes computed are negative. Execution continues if no sky subtraction is required (**reduce.skysub**=FALSE, for instance for SLICER data), but aborts if **reduce.skysub**=TRUE (default).

   Finally, since **reduce.objoffset** may push a sky window beyond the full slit, it will (if positively-sized) be truncated as necessary on the outer edge. It is the user's responsibility to ensure that the full slit does not include inter-order pixels.

   If more than one object trace is present in the slit optimal extraction can be used with a suitable combination of **reduce.slitlength** and **reduce.objoffset** to adapt the slit length and position to cover only one trace and the adjacent sky regions. Obviously the data have to be processed separately for each trace. It is not possible to extract multiple traces with Linear or Average extraction.

- The orders are merged using one of the two possible methods set by the parameter **reduce.merge**.
- The spectrum is corrected to the top of the atmosphere (by correcting for atmospheric extinction if the atmospheric dispersion table is provided) and flux calibrated (see 11.1.22) to physical units (if the instrument response curve is provided).
- Finally, the products are velocity corrected and the `SPECSYS` keyword added. (The correction factor used depends on the **velocity_correction** parameter: "barycentric" uses the value from the `HIERARCH ESO DRS BARYCORR` keyword while "heliocentric" uses the value from `HIERARCH ESO DRS HELICORR`. The parameter defaults to "none".) Note that the application of the correction implies an adjustment of the wavelength values. Therefore if you wish to combine the spectra of the same target, you must interpolate the spectra to a common wavelength range.

11.2.9 uves_obs_redchain

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

- uves_cal_mbias (if no master bias is provided)
- uves_cal_mdark (if no master dark is provided and there are raw dark frames)
- 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 no instrument response is provided and a standard star observation exists)
- 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 the telluric line flat field frames: it generates a MASTER_TFLAT_x frame, divides 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 Workflow technicalities

This section will present some detailed explanations of the UVES Reflex-based workflow.

A.1 Data requirements and optional inputs

In order to run the workflow with a given data set, a number of calibrations and science data must be present. The workflow functions with the concept of science data sets: a set of files that contain a coherent set of science and calibration data.

The minimum coherent data set is composed of:

- A point source science frame.
- A set of bias frames that can be applied to the science, flat and arc lamp frames.
- A set of flat frames that can be applied to the arc lamp and science frames.
- An arc lamp frame for wavelength calibration.
- A continuum lamp spectrum to trace the different orders.
- Static calibrations: Line reference table and extinction coefficient. table

On top of that, there are some optional calibrations that will be used by the workflow *if* they are present:

- A set of darks.
- A standard star taken the same night for flux calibration purposes.

Take into account that from Reflex 2.0 on, it is possible to use different calibrations frames depending on what are they going to be used for. For instance, the associated bias frames to the science could be different than the bias frames associated to the flat. The DataSetChooser shows a nice tree diagram with all the associations.

The conditional execution of the **uves_cal_mdark** and **uves_cal_response** recipes is implemented in subworkflows `Master Dark Creation` and `Instrument Response`.

A workflow accepts science and calibration data, as delivered to PIs in the form of PI-Packs (until October 2011) or downloaded from the archive using the CalSelector tool[15] and organises them into DataSets, where each DataSet contains one science object observation (possibly consisting of several science files) and all associated raw and static calibrations required for a successful data reduction.

For a detailed explanation about how these calibrations are organised and associated, please refer to Section A.3.

[15] http://www.eso.org/sci/archive/calselectorInfo.html

A.2 Overall layout

The workflow structure has several parts:

- Top area. Contains annotations about the workflow and the main parameters to setup the workflow. Note that not *all* the parameters of the workflow are set here: each actor has its own parameters that are set individually.
- Bottom area. This contains all the actors that actually execute or perform a workflow action. It is also composed of several parts:
  - Initialisation actors on the left side. These actors prepare the rest of the workflow to start with the data reduction. It includes setup of required intermediate variables, the data organisation, the data selection and the routing of the data. Please note that the data organisation and selection is performed only once, while the routing is performed once for each data set that has been selected.
  - Recipe execution. The middle part of the bottom area contains the actors that execute pipeline recipes. This is basically where the logic of the data reduction chain is implemented.
  - Closing of the reduction for this data set. The right side of the workflow contains actors that perform the last actions needed after the reduction of a data set. This includes renaming of the final science data, an interactive data display (disabled by default) and some housekeeping of variables.
- Composite actors. Some of the actors are composite of other actors. That means that they contain another workflow inside. To inspect a composite actor, right-click on it, and select `Open Actor`.

A.3 OCA rules

The OCA rules are the mechanism used to Organise, Classify and Associate the data. These rules are stored in a human-readable file which contains several sections, one for each of the required tasks.

The DataOrganiser is the component that makes use of these rules. The parameter `Oca File` specifies the proper OCA rule file to use. It is advised to check that it points to the right location (although the installation procedure should take care of that). Figure A.1 shows the parameter to inspect to check that the OCA rules are ok.

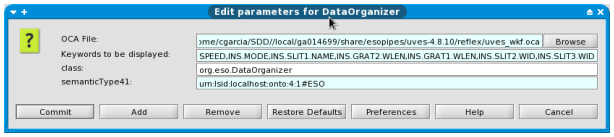


Figure A.1: *Parameter to change for the OCA file, used for classification, grouping and association of data.*

Here we present a brief summary of the meaning of the current OCA rules provided with the workflow:

- The raw data is classified according to the DPR keywords.
- The products of the recipes are classified according to the PRO.CATG keyword.

- The raw biases and darks are grouped by detector binning, detector speed and observation template.
- The raw flats are grouped by detector binning, detector speed, slit id, wavelength grating, slit width and observation template.
- The raw format check data and order reference data are grouped by detector binning, detector speed and wavelength grating.
- The standard star data is grouped by detector binning, detector speed, wavelength grating and star name.
- The science data is grouped by detector binning, detector speed, wavelength grating, slit id, slit width and wavelength grating.
- All the calibrations are directly associated to the science data. If several groups of calibration data match the rules, then the closest in time is chosen.
- The raw biases and raw darks are associated if the detector binning and speed match.
- The raw flats are associated if the detector binning, detector speed, wavelength grating and slit width match.
- The format check data and the order reference data are associated if the detector binning and the wavelength grating match.
- The wavelength calibration is associated if the detector binning, detector speed, wavelength grating and slit width match.
- The standard star is associated if the wavelength grating and the observation day match.
- The master response is associated if the wavelength grating match.

A.4 Workflow fine tuning and hints

We have collected several hints to fine tune and exploit all the capabilities of the workflow.

- Check disk space before starting to reduce the data. The reduction of all the demo data sets will need at least 1 GB. However, at least 5 GB are recommended. This disk space requirement applies to directories pointed to by the `TMP_PRODUCTS` and `END_PRODUCTS_DIR` variables.
- Should the user find data reduction problems, the recipe parameter **debug** allows to increase recipe verbosity. Change this parameter for the recipe of interest.
- The ProductRenamer can be setup to create the desired filenames. Use the `Rename keywords` parameter to change the renaming scheme. It is possible to use keywords from the the header or literal strings (quoted by ”). Currently, there is only one ocurrence in the workflow of the ProductRenamer, just after the science recipe. However, this actor can be placed in several places in the workflow (for instance, to store in the final directory the master flat).
- All the intermediate products created by the workflow are stored in `TMP_PRODUCTS_DIR` directory. The subdirectory structure is shown in Figure A.2. For each recipe instance there is a subdirectory, and inside this there is another subdirectory with the timestamp of the execution time.

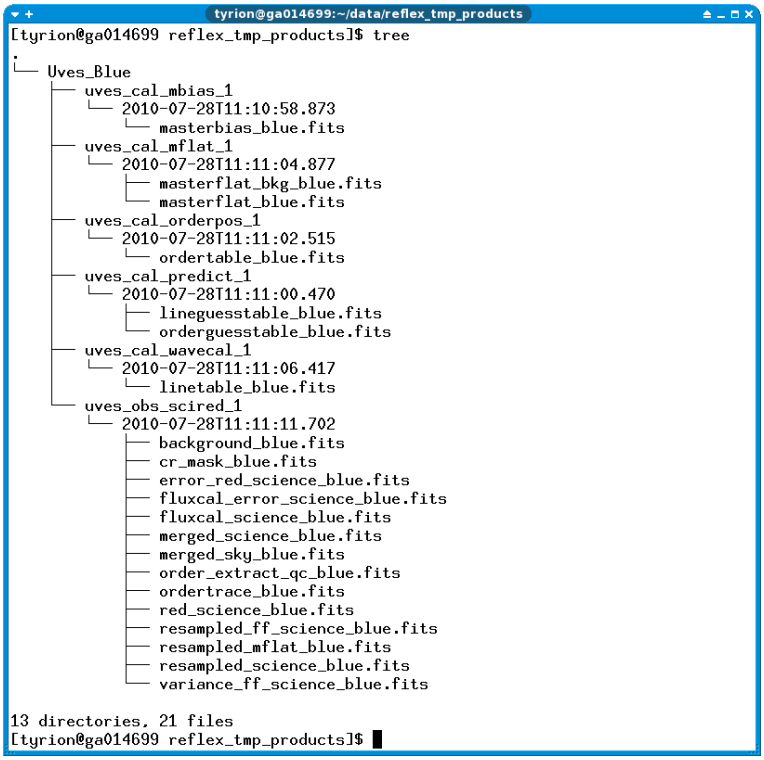


Figure A.2: *Structure of the* `REFLEX_PRODUCTS` *directory.*

B 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 B.1. It is recommended reading through Section B.3 before starting the installation.

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

B.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),

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.

The Reflex workflow is supported only in the following platform:

- Fedora 11

B.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)

For Reflex workflow support it is highly recommended to have installed:

- the Java Development Kit (version 1.6 or newer)
- Python
- pyfits 2.1.1 (http://www.stsci.edu/resources/software_hardware/pyfits)
- wxPython 2.8.10.1 (http://www.wxpython.org/download.php)
- matplotlib 0.99.1.2 (http://matplotlib.sourceforge.net/)
- numpy 1.3.0 (http://numpy.scipy.org/)

These packages are part of the supported platform for Reflex, Fedora 11, and a simple way to install all the needed dependencies is executing as root the following command:

```
# yum install python-matplotlib wxPython pyfits numpy java-1.6.0-openjdk
  python gcc-c++ make gzip tar
```

B.3 Building the UVES pipeline

The UVES pipeline distribution kit contains:

uves-manual-6.6.1.pdf	The UVES pipeline manual
install_pipeline	Install script
qfits-6.2.0.tar.gz	QFITS 6.2.0
cpl-7.3.tar.gz	CPL 7.3
esorex-3.13.6.tar.gz	esorex 3.13.6
gasgano-2.4.8.tar.gz	GASGANO 2.4.8
uves-6.6.1.tar.gz	UVES 6.6.1
uves-calib-6.6.1.tar.gz	UVES static calibration files 6.6.1

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the UVES pipeline 6.6.1 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-6.6.1.tar.gz**

4. Unpack using the following commands:
   gunzip uves-kit-6.6.1.tar.gz tar -xvf uves-kit-6.6.1.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 1.8 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-6.6.1

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

   Note: on recent Mac OS in order to properly install the kit it may be useful to set the following environment variable:

   ```
   export JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/
   ```

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 1.8), 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.3.1 Installing the UVES data demo package

The last version of the data set can be downloaded from:

https://ftp.eso.org/pub/dfs/pipelines/instruments/uves/uves-demo-reflex-4.5.tar.gz.

To extract the data it is required to have the following tools in the path:

- the `gunzip` data compression program,

- a version of the `tar` file-archiving program,
- `wget` or other download manager

To unpack the demo package perform the following steps:

1. Change directory to where you want to store the data demo files. You will need at least 15Gb of free space to unpack and later run the pipeline just one time with one data set. However, at least 50Gb is recommended if several executions are done, though.
2. Retrieve the package with `wget` or other download tool:
3. `wget` *https://ftp.eso.org/pub/dfs/pipelines/instruments/uves/uves-demo-reflex-4.5.tar.gz*
4. Unpack it: `tar xvzf uves-demo-reflex-4.5.tar.gz`

C Abbreviations and acronyms

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

Index

ANSI, 113
ASCII, 113

Background subtraction
  Description, 98

CalibDB, 113
CPL, 113
  web page, 12

Data reduction cascade, 55
Data Reduction Overview, 54
Definition
  ANSI, 113
  ASCII, 113
  CalibDB, 113
  CPL, 113
  DFS, 113
  DMD, 113
  DRS, 113
  EsoRex, 113
  FITS, 113
  FOV, 113
  FPN, 113
  GUI, 113
  OB, 113
  PSO, 113
  QC, 113
  RON, 113
  SOF, 113
  UT, 113
  UVES, 113
  VLT, 113
Description
  Background subtraction, 98
  Efficiency determination, 89
  First guess solution, 93
  Instrument response determination, 101
  Response determination, 89
  Spectrum flux calibration, 90
DFS, 113
DMD, 113
DRS, 113

Efficiency determination, 89
ESO, 113
EsoRex, 113
  Configuration file, 31
  Example
    uves_cal_mbias, 35
    uves_cal_mdark, 36
    uves_cal_orderpos, 37
    uves_cal_predict, 36
    uves_cal_response, 38
    uves_cal_wavecal, 37
    uves_obs_redchain, 33
    uves_obs_scired, 39
  Help, 31
  Introduction, 30
  Parameters listing, 31
  Pipeline recipe confifuration, 31
  Pipeline recipe execution, 32
  Pipeline recipe help, 31
  Pipeline recipe man page, 31
  Pipeline recipes listing, 31
  Set of frames definition, 30
  Syntax, 31
  Usage, 30
  web page, 12

FITS, 113
FOV, 113
FPN, 113

Gasgano
  File classification, 26
  File selection, 26
  Introduction, 20
  Pipeline Recipe execution, 28
  Usage, 26
  web page, 12
GUI, 113

Input data, 54
Instrument response determination, 101

Master frame generation, 90, 93, 98

OB, 113
Order Definition, 97

Phisical Model
  Diagnostic plots, 94
Physical Model, 93
  Parameter corrections, 96
Pipeline
  Cookbook, 19
  Example
    uves_cal_mbias, 35
    uves_cal_mdark, 36
    uves_cal_orderpos, 37
    uves_cal_predict, 36
    uves_cal_response, 38
    uves_cal_wavecal, 37
    uves_obs_redchain, 33
    uves_obs_scired, 39
  Input data, 54
  Overview, 11, 15
  Purposes, 15
  Recent improvements, 16
  Recipe execution, 32
  Recipes
    uves_cal_mbias, 19, 58
    uves_cal_mdark, 19, 59
    uves_cal_mflat, 19, 65, 66
    uves_cal_mflat_combine, 19
    uves_cal_mkmaster, 19, 66
    uves_cal_orderpos, 19, 63
    uves_cal_predict, 19, 61
    uves_cal_response, 19, 71
    uves_cal_tflat, 19
    uves_cal_wavecal, 19, 67
    uves_obs_cd_align, 19
    uves_obs_redchain, 19, 77
    uves_obs_scired, 19, 73
PSO, 113

QC, 113
QFITS
  web page, 12

Raw data, 42
Raw Frames
  Characteristic FITS keywords, 43
  Supported, 43
Reference data
  Atmospheric extinction table, 52
  Line reference table, 52
  Master response curves, 53
  Standard stars flux table, 52
  Table to monitor line intensity, 52
  Table with wavelength points to fit the response, 53
Response determination, 89
RON, 113

Set of frames
  Example
    uves_cal_mbias, 35
    uves_cal_mflat, 35
    uves_cal_orderpos, 34
    uves_cal_predict, 34
    uves_cal_response, 35
    uves_cal_wavecal, 35
    uves_obs_scired, 35
SOF, 113
Spectrum flux calibration, 90

UT, 113
UVES, 17, 113
  Instrument, 17
  Instrument description, 17
  Pipeline publication, 12
  Pipeline User Manual, 12
  Quality Control web page, 12
  Raw data, 42
  Science Verifications web page, 12
  User Calibration Plan, 12
  User Manual, 12
  web page, 12
uves_cal_predict, 93

VLT, 113

Wavelength calibration, 98
  Diagnostic plots
    Residuals, 100
    Resolution, 100