VERY LARGE TELESCOPE

ESPRESSO Pipeline User Manual

VLT-MAN-ESO-19500-XXXX

Issue 1.0
Date 2018-12-10

Prepared: A. Modigliani, D. Sowonsowska, C. Lovis 2018-12-10
Name Date Signature

Approved: P. Ballester
Name Date Signature

Released: M. Peron
Name Date Signature
<table>
<thead>
<tr>
<th>Issue/Rev.</th>
<th>Date</th>
<th>Section/Parag. affected</th>
<th>Reason/Initiation/Documents/Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>10/12/2018</td>
<td></td>
<td>First public release, pipeline version 1.1.0</td>
</tr>
</tbody>
</table>
This page was intentionally left blank
Contents

1 Introduction 11
   1.1 Purpose ................................................. 11
   1.2 Acknowledgements ..................................... 11
   1.3 Scope ................................................. 11
   1.4 Reference documents ................................... 12

2 Overview 13

3 Recent improvements 14

4 ESPRESSO Instrument Description 15
   4.1 Instrument overview ................................... 15
   4.2 Instrument description ................................ 15
   4.3 The Coudé Trains ..................................... 16
   4.4 The Front Ends ....................................... 17
   4.5 The Spectrograph ..................................... 18
   4.6 Observing modes ...................................... 19

5 Quick start 20
   5.1 ESPRESSO pipeline recipes ............................. 20
   5.2 An introduction to Reflex, Gasgano and EsoRex ...... 20
      5.2.1 Using Reflex ...................................... 21
      5.2.2 Example of data reduction using the Reflex-based ESPRESSO workflow .. 22
      5.2.3 Using EsoRex .................................... 26
   5.3 Example of data reduction using EsoRex .......... 29

6 Known problems 37

7 Instrument Data Description 38
   7.1 BIAS frames ........................................... 39
   7.2 DARK frames .......................................... 39
   7.3 Detector Flat-field (LED) frames .................... 39
7.4 ORDERDEF frames .......................................................... 40
7.5 FLAT frames ............................................................... 40
7.6 FP-FP frames ............................................................... 40
7.7 THAR-FP frames ........................................................... 40
7.8 FP-THAR frames ........................................................... 40
7.9 LFC-FP frames ............................................................. 40
7.10 FP-LFC frames ............................................................ 41
7.11 Contamination by simultaneous reference frames ................. 41
7.12 Fiber-to-fiber Relative Efficiency frames ................................. 41
7.13 Spectrophotometric Calibration (FLUX) frames ....................... 41
7.14 SCIENCE frames ........................................................ 41

8 Static Calibration Data ....................................................... 42
8.1 CCD Geometry Table ..................................................... 42
8.2 Instrument Configuration Table ......................................... 42
8.3 Led Flat Field gain windows Table ..................................... 42
8.4 Wavelength Matrix Images .............................................. 42
8.5 DLL Matrix Images ....................................................... 42
8.6 THAR line tables .......................................................... 42
8.7 Flux Standard Star tables ................................................. 43
8.8 Extinction Table ........................................................... 43
8.9 Flux Template Tables ..................................................... 43
8.10 CCF Template Tables .................................................... 43

9 Data Reduction ............................................................... 44

10 Pipeline Recipes Interfaces ............................................. 47
10.1 espdr_mbias ................................................................. 47
  10.1.1 Input ................................................................. 47
  10.1.2 Output ............................................................... 47
  10.1.3 Quality control ...................................................... 47
  10.1.4 Parameters .......................................................... 48
10.2 espdr_mdark ................................................................. 48
  10.2.1 Input ................................................................. 48
  10.2.2 Output ............................................................... 48
  10.2.3 Quality control .................................................... 49
  10.2.4 Parameters ......................................................... 49
10.3 espdr_led_ff .............................................................. 49
  10.3.1 Input ................................................................. 49
  10.3.2 Output ............................................................... 49
  10.3.3 Quality control .................................................... 49
  10.3.4 Parameters ......................................................... 50
10.4 espdr_orderdef .......................................................... 50
  10.4.1 Input ................................................................. 50
  10.4.2 Output ............................................................... 50
  10.4.3 Quality control .................................................... 50
  10.4.4 Parameters ......................................................... 51
10.5 espdr_mflat .............................................................. 51
  10.5.1 Input ................................................................. 51
  10.5.2 Output ............................................................... 51
  10.5.3 Quality control .................................................... 52
  10.5.4 Parameters ......................................................... 52
10.6 espdr_wave_FP ........................................................... 52
  10.6.1 Input ................................................................. 52
  10.6.2 Output ............................................................... 52
  10.6.3 Quality control .................................................... 53
  10.6.4 Parameters ......................................................... 53
10.7 espdr_wave_THAR ....................................................... 53
  10.7.1 Input ................................................................. 53
  10.7.2 Output ............................................................... 54
  10.7.3 Quality control .................................................... 54
  10.7.4 Parameters ......................................................... 55
10.8 espdr_wave_LFC .................................................. 55
  10.8.1 Input .......................................................... 55
  10.8.2 Output ........................................................ 55
  10.8.3 Quality control ............................................. 55
  10.8.4 Parameters .................................................. 55
10.9  espdr_contam .................................................... 55
  10.9.1 Input .......................................................... 55
  10.9.2 Output ........................................................ 56
  10.9.3 Quality control ............................................. 56
  10.9.4 Parameters .................................................. 56
10.10 espdr_eff_ab ..................................................... 56
  10.10.1 Input .......................................................... 56
  10.10.2 Output ........................................................ 57
  10.10.3 Quality control ............................................. 57
  10.10.4 Parameters .................................................. 57
10.11 espdr_cal_flux ................................................... 58
  10.11.1 Input .......................................................... 58
  10.11.2 Output ........................................................ 58
  10.11.3 Quality control ............................................. 58
  10.11.4 Parameters .................................................. 59
10.12 espdr_sci_red .................................................... 59
  10.12.1 Input .......................................................... 59
  10.12.2 Output ........................................................ 60
  10.12.3 Quality control ............................................. 60
  10.12.4 Parameters .................................................. 61

11 Algorithms and recipe details .................................. 62
  11.1 Algorithms ...................................................... 62
    11.1.1 Master frame creation using a kappa-sigma-clipped mean/median .......... 62
    11.1.2 Order definition ........................................... 62
    11.1.3 Background subtraction .................................... 62
11.1.4 Spectrum Extraction ................................................................. 63
11.1.5 Flat-Fielding ................................................................. 64
11.1.6 Wavelength Calibration ....................................................... 64
11.1.7 Instrumental Drift Measurement ................................................. 65
11.1.8 Sky Subtraction .............................................................. 66
11.1.9 Barycentric Correction .............................................................. 66
11.1.10 Order Rebinning and Merging .................................................. 68
11.1.11 Radial Velocity Computation .................................................. 68
11.1.12 Error and Bad Pixel Propagation ................................................. 69
11.2 Recipes ................................................................. 70
11.2.1 mbias ................................................................. 70
11.2.2 mdark ................................................................. 70
11.2.3 led_ff ................................................................. 70
11.2.4 orderdef ............................................................. 71
11.2.5 mflat ................................................................. 71
11.2.6 cal_contam ............................................................. 72
11.2.7 cal_eff_ab ............................................................. 72
11.2.8 wave_FP ............................................................. 72
11.2.9 wave_THAR .......................................................... 73
11.2.10 wave_LFC .......................................................... 73
11.2.11 cal_flux ............................................................. 74
11.2.12 sci_red ............................................................. 74

A Installation ............................................................... 76
A.1 Supported platforms .......................................................... 76
A.2 Building the ESPRESSO pipeline .............................................. 76
A.2.1 Requirements .......................................................... 76
A.2.2 Compiling and installing the ESPRESSO pipeline .............. 77
1 Introduction

1.1 Purpose

The ESPRESSO pipeline is a subsystem of the VLT Data Flow System (DFS). Its target user is ESO Data Products Department (DPD) 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 ESPRESSO 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 ESPRESSO data reduction sequence with the ESPRESSO pipeline.

This manual is a complete description of the data reduction recipes implemented by the the ESPRESSO pipeline, reflecting the status of the ESPRESSO pipeline as of Dec 10, 2018 (version 1.1.0).

1.2 Acknowledgements

The ESPRESSO pipeline has been designed, implemented and developed by the Geneva Observatory. We are particularly grateful to the responsible for the data reduction: Christophe Lovis, Danuta Sownowska and Alex Segovia for their contributions and support.

Andrea Modigliani, from ESO, provided the Reflex workflow, DFS support and part of the documentation. We thank also Andrea Mehner and Richard Anderson for providing useful feedback to improve the reflex workflow.

1.3 Scope

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

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

Additional information on the Common Pipeline Library (CPL) and ESOREX can be found respectively at [4], [5]. A description of the instrument is in [6]. The ESPRESSO instrument user manual is in [7].
1.4 Reference documents

   www.eso.org/pipelines
[2] Current pipeline status  
   www.eso.org/observing/dfo/quality/pipeline-status.html
[6] ESPRESSO home page  
   www.eso.org/sci/facilities/paranal/instruments/espresso
   www.eso.org/sci/facilities/paranal/instruments/espresso/doc
[8] Reflex ESPRESSO Tutorial  
   www.eso.org/pipelines
2 Overview

In collaboration with instrument consortia, the Pipeline Systems Department (PSD) of the Software Development Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

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

**Master calibration product creation:** pipelines are used to produce master calibration products (e.g., combined bias frames, super-flats, wavelength dispersion solutions).

**Science product creation:** using pipeline-generated master calibration products, science products are produced for the supported instrument modes (e.g., combined ISAAC jitter stacks; bias-corrected, flat-fielded FORS images, wavelength-calibrated UVES spectra). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

Instrument pipelines consist of a set of data processing modules that can be called from the command line, from the automatic data management tools available on Paranal or from Gasgano.

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* and *EsoRex* [5] both included in the pipeline distribution (see Appendix A, page 76). These applications can also be downloaded separately from [www.eso.org/gasgano](http://www.eso.org/gasgano) and [www.eso.org/cpl/esorex.html](http://www.eso.org/cpl/esorex.html). An illustrated introduction to Gasgano is provided in the "Quick Start" section of this manual (see page 20).

The ESPRESSO instrument and the different types of ESPRESSO raw frames and auxiliary data are described in Sections 4, 7, and 8.

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

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

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

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

In Appendix A the installation of the ESPRESSO pipeline recipes is described.
3 Recent improvements

This is the first release of the ESPRESSO pipeline.
4 ESPRESSO Instrument Description

ESPResso is the Echelle SPectrograph for Rocky Exoplanets and Stable Spectroscopic Observations installed at the incoherent combined Coudé facility of the VLT. It is an ultra-stable fibre-fed échelle high-resolution spectrograph (R 140,000, 190,000, or 70,000) which collects the light from either a single UT or the four UTs simultaneously via the so-called UT Coudé trains. The whole system is built to reach the ultimate radial-velocity precision of 10 cm/s over a timespan of 10 years in single UT mode.

In this chapter a brief description of the ESPRESSO instrument is given. A more complete documentation can be found in the ESPRESSO User Manual, downloadable from www.eso.org/sci/facilities/paranal/instruments/espresso.

4.1 Instrument overview

ESPResso is the ESO/VLT high-resolution spectrograph to measure precise radial velocities on a long timespan with the main scientific aim of detecting and characterising Earth twins in the habitable zone of solar-like stars.

ESPResso is a highly-stabilized fibre-fed échelle spectrograph that can be fed with light from either a single or the four Unit Telescopes (UTs) simultaneously. The instrument is installed at the incoherent combined Coudé focus (ICCF) of the VLT. The light from an astronomical source is redirected from the telescopes to the detectors through three components of the ICCF facility: the UT Coudé trains, the front end units, and the spectrograph itself. The Coudé Trains (CT) bring the light from each telescope to the Combined Coudé Lab (CCL) through 13 optical elements, including mirrors, lenses, and prisms. The four Front Ends (one for each UT) receive the light from the CTs and feed the spectrograph entrance fibres. The Fibre Link transports the light from the Front Ends to the vacuum vessel. The latter is thermally stabilized at the mK level. The light is then going through the optical components of the spectrograph and split up into a red and a blue spectrum which are recorded on the corresponding science detectors.

The spectrograph is fed by two fibres, one for the target and the other one for simultaneous calibration (either the sky or a simultaneous reference: Fabry-Pérot, Laser Frequency Comb, or Thorium-Argon lamp). The light from the two fibres is recorded onto a blue (380-525nm) and a red (525-788nm) CCD mosaic. ESPRESSO can operate in three main instrument configurations: High Resolution 1-UT (HR), Ultra High-Resolution 1-UT (UHR) and Medium Resolution 4-UT (MR). The main characteristics of these modes are summarised below (for more details on the characteristics of the instrument, see the Instrument Description).

4.2 Instrument description

ESPResso is the first instrument of a new VLT facility that can host instrumentation at the so-called Incoherent Combined Coudé Focus (ICCF). As such, in order to describe the instrument, one needs to understand the three components through which the light travels from the telescope to the detectors. These are: the UT Coudé Trains, the Front Ends, and the Spectrograph itself. Each of these components are described below. For a more detailed description, we point the interested reader to the ESPRESSO user manual.
Table 4.1.0: The EXPRESSO pipeline actors and their contents

<table>
<thead>
<tr>
<th></th>
<th>HR(1-UT)</th>
<th>UHR(1-UT)</th>
<th>MR(4-UT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelength range</td>
<td>380-788 nm</td>
<td>380-788 nm</td>
<td>380-788 nm</td>
</tr>
<tr>
<td>Resolving power (median)</td>
<td>140,000</td>
<td>190,000</td>
<td>70,000</td>
</tr>
<tr>
<td>Aperture on sky</td>
<td>1&quot;.0</td>
<td>0&quot;.5</td>
<td>4x1&quot;.0</td>
</tr>
<tr>
<td>Global efficiency @ 550nm</td>
<td>9%</td>
<td>4%</td>
<td>9%</td>
</tr>
<tr>
<td>RV precision (requirement)</td>
<td>&lt; 10 cm/s</td>
<td>&lt; 5 m/s</td>
<td>&lt; 5 m/s</td>
</tr>
<tr>
<td>Limiting V-band magnitude</td>
<td>≈ 17</td>
<td>≈ 16</td>
<td>≈ 20</td>
</tr>
<tr>
<td>Binning</td>
<td>1x1, 2x1</td>
<td>1x1</td>
<td>4x2, 8x4</td>
</tr>
<tr>
<td>Spectral sampling (average)</td>
<td>4.5 px</td>
<td>2.5 px</td>
<td>5.5 px (binned x2)</td>
</tr>
<tr>
<td>Spectral sampling per slice</td>
<td>9.0(4.5) px</td>
<td>5.0 px</td>
<td>5.5 px (binned x4)</td>
</tr>
<tr>
<td>Number of slices</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 4.1.1: An view of ESPRESSO

4.3 The Coudé Trains

Distances between each UT and the Combined Coudé Laboratory (CCL) range between 48 m for UT2 and 69 m for UT1. A trade-off analysis between solutions based on mirrors, prisms, lenses, and fibres, points towards a full-optics solution. The chosen design with the position of the 11 optical elements is shown below in the second figure. The Coudé train picks up the light through a prism at the level of the Nasmyth-B platform and routes the beam through the UT mechanical structure down to the UT Coudé room, and farther to the CCL along the existing incoherent light ducts. The four trains relay a field of 17 arcsec around the acquired object to the CCL. The selected concept to convey the light of the telescope from the Nasmyth-B focus to the entrance of the tunnel in the Coudé room (CR) below each UT is based on a set of 6 prisms (with some power). The light is directed from the UT’s Coudé room towards the CCL using 2 large lenses. The beams from the four UTs meet in the CCL, where mode selection and beam conditioning is performed by the fore-optics of the Front-End sub-system.
4.4 The Front Ends

The Front-End transports the beam received from the Coudé, once corrected for atmospheric dispersion by the ADC, to the common focal plane where the pickups for the spectrograph fiber feeds are located. While performing such a beam conditioning, the Front-End can apply pupil and field stabilizations. These are achieved via two independent control loops each composed of a technical camera and a tip-tilt stage. Another dedicated stage delivers a focusing function. In addition, the Front-End handles the injection of the calibration light, prepared in the Calibration Unit, into the fibers and then into the spectrograph. As calibration sources, ESPRESSO is equipped with a laser frequency comb (LFC), two ThAr lamps (one for simultaneous reference and one for calibration) and a calibration light is foreseen to be composed of two ThAr, one for simultaneous Fabry-Pérot calibration. A toggling mechanism handles the selection between the possible observing modes in a passive way.

Link sub-system relays the light from the Front-End to the spectrograph and forms the spectrograph pseudo-slit inside the vacuum vessel. The 1-UT mode uses a bundle of 2 octagonal fibres each, one for the object and one for the sky or simultaneous reference. In the high-resolution (singleHR) mode, the fibre has a core of 140 µm, equivalent to 1 arcsec on the sky; in the ultra-high-resolution (singleUHR) mode, the fibre core is 70 µm and the covered field of view is 0.5 arcsec. The fibre entrances are organized in pickup heads that are moved to the focal plane of the Front End when the specific bundle of the specific mode is selected. In the 4-UT mode (multiMR), four object fibres and four sky/reference fibres are fed simultaneously by the four telescopes. The four object fibres will finally feed a single square 280 µm object fibre, while the four sky/reference fibres will feed a single square 280 µm sky/reference fibre. In the 4-UT mode, the spectrograph will see a pseudo slit of four fibre images, although they will be square and twice as wide as the 1-UT fibres. Another essential task performed by the Fibre-link sub-system is the light scrambling. The use of a double-scrambling optical system will ensure both scrambling of the near field and far field of the light beam. A high scrambling gain, which is crucial to obtain the required RV precision in the 1-UT mode is achieved by the use of octagonal fibres (Chazelas et al. 2011).
4.5 The Spectrograph

The spectrograph optics are mounted in a 3-dimensional optical bench specifically designed to keep the optical system within the thermo-mechanical tolerances required for high-precision radial velocity measurements. The bench is mounted in a vacuum vessel in which 10^{-5} mbar class vacuum is maintained during the entire duty cycle of the instrument. The temperature at the level of the optical system is required to be stable at the mK level in order to avoid both short-term drift and long-term mechanical instabilities. Such an ambitious requirement is obtained by locating the spectrograph in a multi-shell active thermal enclosure system. Each shell improves the temperature stability by a factor of 10, thus getting from typically Kelvin-level variations in the CCL down to 1 mK stability inside the vacuum vessel and on the optical bench.

At the entrance of the spectrograph, an anamorphic pupil slicing unit (APSU) shapes the beam in order to compress the beam in cross-dispersion direction but not in main-dispersion direction, where high resolving power needs to be achieved. In the latter direction, however, the pupil is sliced and superimposed on the echelle grating to minimize its size. The rectangular white-pupil is then re-imaged and compressed by the anamorphic VPH grism. Given the wide spectral range and the required efficiency, two large 90x90 mm CCD detectors are required to record the full spectrum. Therefore, a dichroic beam splitter separates the beam in a blue and a red channel which in turn allows to optimize each spectroscopic arm for image quality and optical efficiency.

The cross-disperser has the function of separating the dispersed spectrum in all its spectral orders. In addition, an anamorphism is re-introduced to make the pupil square and to compress the order width in cross-dispersion direction, such that the inter-order space is maximized.

Figure 4.5.1: ESPRESSO: opto-mechanical path (left) and optical path (right).
4.6 Observing modes

The instrument can be operated in three observing modes: High Resolution 1-UT (HR), Ultra High-Resolution 1-UT (UHR), and Medium Resolution 4-UT (MR). The main characteristics of the corresponding five instrument configurations are summarised below:

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Detector readout mode</th>
<th>CCD pixel binning</th>
<th>Combined readout, transfer &amp; warming time</th>
<th>RON (Blue detector)</th>
<th>RON (Red detector)</th>
<th>CDNAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGLE_HR1</td>
<td>Spectroscopy and RV monitoring of bright stars (&gt;15-12) with 1 UT</td>
<td>FAST</td>
<td>1x1</td>
<td>45 s</td>
<td>5 s/pix</td>
<td>5 s/pix</td>
</tr>
<tr>
<td>SINGLE_UHR11</td>
<td>Very high-resolution spectroscopy with 1 UT</td>
<td>FAST</td>
<td>1x1</td>
<td>45 s</td>
<td>8 s/pix</td>
<td>8 s/pix</td>
</tr>
<tr>
<td>SINGLE_HR21</td>
<td>Spectroscopy and RV monitoring of faint targets (&gt;15-12) with 1 UT</td>
<td>SLOW</td>
<td>2x1</td>
<td>60 s</td>
<td>3 s/pix</td>
<td>2 s/pix</td>
</tr>
<tr>
<td>MULTI_M442 [offered starting in P120]</td>
<td>Spectroscopy of faint targets with 4 UTs</td>
<td>SLOW</td>
<td>4x2</td>
<td>41 s</td>
<td>3 s/pix</td>
<td>2 s/pix</td>
</tr>
<tr>
<td>MULTI_M442 [offered starting in P120]</td>
<td>Spectroscopy of very faint targets with 4 UTs</td>
<td>SLOW</td>
<td>8x4</td>
<td>38 s</td>
<td>3 s/pix</td>
<td>2 s/pix</td>
</tr>
</tbody>
</table>

Figure 4.6.1: ESPRESSO observing modes.
5 Quick start

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

5.1 ESPRESSO pipeline recipes

The current ESPRESSO pipeline is based on a set of 6 stand-alone recipes involved in the data reduction cascade:

- `espdr_mbias` Creates the master bias frame
- `espdr_mdark` Creates the master dark & hot pixel mask
- `espdr_led_ff` Computes the mean gain and detect the bad pixels
- `espdr_orderdef` Defines the orders on the CCD
- `espdr_mflat` Creates the master flat
- `espdr_wave_FP` Wavelength FP calibration (input FP,FP)
- `espdr_wave_THAR` Wavelength calibration (input FP,THAR or THAR,FP)
- `espdr_wave_LFC` Wavelength calibration with LFC
- `espdr_cal_contam` Generate a contamination frame and check contamination
- `espdr_cal_eff_ab` Compute the relative efficiency between sky and
- `espdr_cal_flux` Measure the absolute efficiency curve
- `espdr_sci_red` Performs science reduction

Other three stand-alone recipes are also provided, used during commissioning:

- `espdr_compu_drift` Measures instrumental drift on wavelength calibration
- `espdr_wave_THAR_THAR` S2D extraction of THAR,THAR frames
- `espdr_wave_LFC_LFC` S2D extraction of LFC,LFC frames

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**.

This manual does not cover the installation of **Reflex**. Please refer to [8] 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:

```bash
user@host# esoreflex &
```

**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 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 5.2.2. 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 [8].
Figure 5.2.1: _Fresh Reflex canvas._

### 5.2.2 Example of data reduction using the Reflex-based ESPRESSO workflow
Figure 5.2.2: Parameters of a recipe actor.

Figure 5.2.3: ESPRESSO workflow general layout.
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 ESPRESSO demo data set supplied with the Reflex 2.8 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:
   
   esoreflex &

   The empty Reflex canvas as shown in Figure 5.2.1 will appear.

2. Now open the ESPRESSO workflow by clicking on File -> Open File, selecting first espdr-1.1.0 and then the file espdr.xml in the file browser. You will be presented with the workflow canvas shown in Figure 5.2.3. Note that the workflow will appear as a canvas in a new window.

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 data¹, 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.

¹If 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.
Figure 5.2.5: The interactive pop-up window for the Science Data Reduction actor and ESPRESSO pipeline recipe espdr_sci_red.
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 5.2.4) that lists the DataSets along with the values of a selection of useful header keywords\(^2\). 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 reduction of the current DataSet finishes, a pop-up window will appear showing the directory where the final products have been saved.

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

11. 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.

5.2.3 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. On the other side, EsoRex doesn’t offer all the facilities available with Gasgano, and the user must classify and associate the data using the information contained in the FITS header keywords (see Section 7, page 38). The user should also take care of defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

The set-of-frames: Each pipeline recipe is run on a set of input FITS data files. When using EsoRex the filenames must be listed together with their DO category\(^3\) in an ASCII file, the set-of-frames (SOF), that is required when launching a recipe.\(^4\)

Here is an example of SOF, valid for the espdr_orderdef recipe

```
/file_path/ESPRESSO_bad_pixels_SINGLEUHR.fits BAD_PIXEL_MASK
/file_path/ESPRESSO_1x1_CCD_geom_config.fits CCD_GEOM
/file_path/ESPRESSO_hot_pixels_SINGLEHR11.fits HOT_PIXEL_MASK
```

\(^2\)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.

\(^3\)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.

\(^4\)The set-of-frames corresponds to the Input Frames panel of the Gasgano recipe execution window.
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 ESPRESSO 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 `espdr_orderdef` will treat the frame `/file_path/ESPRE.2018-09-02T10:24:23.827.fits` as a ORDERDEF_A, the frame `/file_path/ESPRESSO_master_bias_res.fits` as a MASTER_BIAS_RES, 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 band and pre-optic, and that the appropriate calibration files have been specified.

The reason of this lack of control is that the ESPRESSO recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them.

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

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

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

To get more information on how to customise ESOREX (see also [5]) run the command:

```
esorex -help
```

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

```
esorex -create-config
```

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

```
esorex -recipes
```

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

```
esorex -params recipe_name
```

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

```
esorex -help recipe_name
```

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

```
esorex -man-page recipe_name
```

**Recipe configuration:** Each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe. The configuration files are normally generated.

---

5The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window.
in the directory $HOME/.esorex, and have the same name as the recipe to which they are related, with the filename extension .rc. For instance, the recipe espdr_sci_red has its EsoRex generated configuration file named espdr_sci_red.rc, and is generated with the command:

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

The recipe configuration file is generated with the command:

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

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

```
# --rv_center
# Approximate RV.
espdr.espdr_sci_red.rv_center=19.0
```

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

The recipes provided by the ESPRESSO pipeline are designed to implement a cascade of macro data reduction steps, each controlled by its own parameters. For this reason and to prevent parameter name clashes we specify as parameter prefix not only the instrument name but also the name of the step they refer to. Shorter parameter aliases are made available for use on the command line.

The command

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

generates a default configuration file recipe_name.rc in the directory $HOME/.esorex.

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

```
esorex - -recipe-config=my_alternative_recipe_config
```

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

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

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

```
esorex espdr_sci_red espdr_sci_red.sof
```

The recipe parameters can be modified either by editing directly the used configuration file, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name and before the SOF name, and they will supersede the system defaults and/or the configuration file settings. For instance, to set the espdr_sci_red recipe rv_center parameter to 3.0, the following should be typed:

```
esorex espdr_sci_red - -rv_center=3.0 espdr_sci_red.sof
```

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

---

6If a number of recipe parameters are specified on the command line, the given values will be used in the created configuration file.
5.3 Example of data reduction using EsoRex

A simple, typical data reduction procedure is described here.\textsuperscript{7}

We suggest the user to organize his data per ins.mode, detector binning, data type. Dark frames may be grouped per detector DIT. The detector DIT is given by the value of the FITS keyword DET DIT\textsuperscript{8}. The sinstrument mode is indicated by the value of the FITS keyword INS MODE. The detector bin along X is indicated by the value of DET BINX. In the examples below we suppose the user has data acquired in mode SINGLEHR11. In the following examples /path_raw/ indicates the full path to the source tree directory containing raw data.

Bias Frames: those frames are characterized by DPR.TYPE='BIAS',

\begin{verbatim}
/path_raw/ESPRE.2018-09-02T11:00:27.443.fits BIAS
/path_raw/ESPRE.2018-09-02T11:01:34.278.fits BIAS
/path_raw/ESPRE.2018-09-02T11:02:41.057.fits BIAS
/path_raw/ESPRE.2018-09-02T11:03:49.938.fits BIAS
/path_raw/ESPRE.2018-09-02T11:04:58.840.fits BIAS
/path_raw/ESPRE.2018-09-02T11:06:05.651.fits BIAS
/path_raw/ESPRE.2018-09-02T11:07:14.526.fits BIAS
/path_raw/ESPRE.2018-09-02T11:08:21.291.fits BIAS
/path_raw/ESPRE.2018-09-02T11:09:28.075.fits BIAS
/path_raw/ESPRE.2018-09-02T11:10:34.907.fits BIAS
/path_cdb/ESPRESSO_2x1_CCD_geom_config.fits CCD_GEOM
/path_cdb/ESPRESSO_SINGLEHR_2x1_inst_config_1slice.fits INST_CONFIG
\end{verbatim}

Dark Frames: those frames are characterized by DPR.TYPE='DARK',

\begin{verbatim}
/path_raw/ESPRE.2018-09-02T11:00:27.443.fits DARK
/path_raw/ESPRE.2018-09-02T11:01:34.278.fits DARK
/path_raw/ESPRE.2018-09-02T11:02:41.057.fits DARK
/path_raw/ESPRE.2018-09-02T11:03:49.938.fits DARK
/path_raw/ESPRE.2018-09-02T11:04:58.840.fits DARK
/path_raw/ESPRE.2018-09-02T11:06:05.651.fits DARK
/path_raw/ESPRE.2018-09-02T11:07:14.526.fits DARK
/path_raw/ESPRE.2018-09-02T11:08:21.291.fits DARK
/path_raw/ESPRE.2018-09-02T11:09:28.075.fits DARK
/path_raw/ESPRE.2018-09-02T11:10:34.907.fits DARK
/path_cdb/ESPRESSO_2x1_CCD_geom_config.fits CCD_GEOM
/path_cdb/ESPRESSO_SINGLEHR_2x1_inst_config_1slice.fits INST_CONFIG
\end{verbatim}

LED flat field frames: those frames are characterized by DPR.TYPE='LED'

\begin{verbatim}
/path_raw/ESPRE.2018-09-02T11:00:27.443.fits LED
/path_raw/ESPRE.2018-09-02T11:01:34.278.fits LED
/path_raw/ESPRE.2018-09-02T11:02:41.057.fits LED
/path_raw/ESPRE.2018-09-02T11:03:49.938.fits LED
/path_raw/ESPRE.2018-09-02T11:04:58.840.fits LED
/path_raw/ESPRE.2018-09-02T11:06:05.651.fits LED
/path_raw/ESPRE.2018-09-02T11:07:14.526.fits LED
/path_raw/ESPRE.2018-09-02T11:08:21.291.fits LED
/path_raw/ESPRE.2018-09-02T11:09:28.075.fits LED
/path_raw/ESPRE.2018-09-02T11:10:34.907.fits LED
/path_cdb/ESPRESSO_2x1_CCD_geom_config.fits CCD_GEOM
/path_cdb/ESPRESSO_SINGLEHR_2x1_inst_config_1slice.fits INST_CONFIG
/path_cdb/ESPRESSO_MASTER_BIAS_RES.fits MASTER_BIAS_RES
\end{verbatim}

\textsuperscript{7}The procedure using \textit{Gasgano} is conceptually identical.
\textsuperscript{8}We omit here the prefix HIERARCH ESO
ORDERDEF flat field frames: those frames are characterized by DPR.TYPE='ORDERDEF,LAMP,OFF', 'ORDERDEF,OFF,LAMP'

<table>
<thead>
<tr>
<th>Date</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
</table>

Flat field frames: those frames are characterized by DPR.TYPE='FLAT,LAMP,OFF', 'FLAT,OFF,LAMP'

<table>
<thead>
<tr>
<th>Date</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-09-02T11:14:51.979</td>
<td>/path_raw/ESPRE.2018-09-02T11:14:51.979.fits</td>
<td>FLAT_A</td>
</tr>
<tr>
<td>2018-09-02T11:16:00.163</td>
<td>/path_raw/ESPRE.2018-09-02T11:16:00.163.fits</td>
<td>FLAT_A</td>
</tr>
<tr>
<td>2018-09-02T11:17:08.344</td>
<td>/path_raw/ESPRE.2018-09-02T11:17:08.344.fits</td>
<td>FLAT_A</td>
</tr>
<tr>
<td>2018-09-02T11:20:32.726</td>
<td>/path_raw/ESPRE.2018-09-02T11:20:32.726.fits</td>
<td>FLAT_A</td>
</tr>
<tr>
<td>2018-09-02T11:21:40.874</td>
<td>/path_raw/ESPRE.2018-09-02T11:21:40.874.fits</td>
<td>FLAT_A</td>
</tr>
<tr>
<td>2018-09-02T11:22:49.050</td>
<td>/path_raw/ESPRE.2018-09-02T11:22:49.050.fits</td>
<td>FLAT_A</td>
</tr>
<tr>
<td>2018-09-02T11:23:57.269</td>
<td>/path_raw/ESPRE.2018-09-02T11:23:57.269.fits</td>
<td>FLAT_A</td>
</tr>
<tr>
<td>2018-09-02T11:25:05.370</td>
<td>/path_raw/ESPRE.2018-09-02T11:25:05.370.fits</td>
<td>FLAT_A</td>
</tr>
</tbody>
</table>
The text contains a list of file paths and associated FITS files. Here are the files mentioned:

- `/path_raw/ESPRE.2018-09-02T11:27:36.335.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:28:44.370.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:29:52.575.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:31:00.764.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:32:08.905.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:33:17.145.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:34:25.326.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:35:33.539.fits FLAT_B`
- `/path_raw/ESPRE.2018-09-02T11:36:41.743.fits FLAT_B`

- `/path_cdb/ESPRE.2018-09-01T18:33:24.369.fits FP_FP`
- `/path_cdb/ESPRESSO_2x1_CCD_geom_config.fits CCD_GEOM`
- `/path_cdb/ESPRE.2018-09-01T18:30:05.522.fits THAR_FP`

Other files are mentioned, but the full list is not shown here.
/path_cdb/ESPRESSO_hot_pixels_SINGLEHR21.fits HOT_PIXEL_MASK
/path_cdb/ESPRESSO_bad_pixels_SINGLEHR21.fits BAD_PIXEL_MASK
/path_cdb/ESPRESSO_ORDER_TABLE_A.fits ORDER_TABLE_A
/path_cdb/ESPRESSO_ORDER_TABLE_B.fits ORDER_TABLE_B
/path_cdb/ESPRESSO_BLAZE_A.fits BLAZE_A
/path_cdb/ESPRESSO_BLAZE_B.fits BLAZE_B
/path_cdb/ESPRESSO_FLAT_A.fits FSPECTRUM_A
/path_cdb/ESPRESSO_FLAT_B.fits FSPECTRUM_B
/path_cdb/ESPRESSO_ORDER_PROFILE_A.fits ORDER_PROFILE_A
/path_cdb/ESPRESSO_ORDER_PROFILE_B.fits ORDER_PROFILE_B
/path_cdb/ESPRESSO_FP_SEARCHED_LINE_TABLE_A.fits FP_SEARCHED_LINE_TABLE_FP_FP_A
/path_cdb/ESPRESSO_FP_SEARCHED_LINE_TABLE_A.fits FP_SEARCHED_LINE_TABLE_FP_FP_A
/path_cdb/ESPRESSO_SINGLEHR_1x1_TH_REF_LINE_TABLE_A.fits REF_LINE_TABLE_A
/path_cdb/ESPRESSO_SINGLEHR_1x1_TH_REF_LINE_TABLE_B.fits REF_LINE_TABLE_B
/path_cdb/ESPRESSO_S2D_BLAZE_FP_FP_A.fits S2D_BLAZE_FP_FP_A
/path_cdb/ESPRESSO_S2D_BLAZE_FP_FP_A.fits S2D_BLAZE_FP_FP_A
/path_cdb/ESPRESSO_S2D_BLAZE_FP_FP_B.fits S2D_BLAZE_FP_FP_B
/path_cdb/ESPRESSO_S2D_BLAZE_FP_FP_B.fits S2D_BLAZE_FP_FP_B
/path_cdb/ESPRESSO_SINGLEHR_1x1_STATIC_DLL_MATRIX_A.fits STATIC_DLL_MATRIX_A
/path_cdb/ESPRESSO_SINGLEHR_1x1_STATIC_DLL_MATRIX_B.fits STATIC_DLL_MATRIX_B
/path_cdb/ESPRESSO_SINGLEHR_1x1_STATIC_WAVE_MATRIX_A.fits STATIC_WAVE_MATRIX_A
/path_cdb/ESPRESSO_SINGLEHR_1x1_STATIC_WAVE_MATRIX_B.fits STATIC_WAVE_MATRIX_B

and

/path_raw/ESPRE.2018-09-01T18:31:46.008.fits THAR_FP
/path_cdb/ESPRESSO_2x1_CCD_geom_config.fits CCD_GEOM
/path_cdb/ESPRESSO_SINGLEHR_2x1_inst_config_1slice.fits INST_CONFIG
/path_cdb/ESPRESSO_master_bias_res.fits MASTER_BIAS_RES
/path_cdb/ESPRESSO_hot_pixels_SINGLEHR21.fits HOT_PIXEL_MASK
/path_cdb/ESPRESSO_bad_pixels_SINGLEHR21.fits BAD_PIXEL_MASK
/path_cdb/ESPRESSO_ORDER_TABLE_A.fits ORDER_TABLE_A
/path_cdb/ESPRESSO_ORDER_TABLE_B.fits ORDER_TABLE_B
/path_cdb/ESPRESSO_BLAZE_A.fits BLAZE_A
/path_cdb/ESPRESSO_BLAZE_B.fits BLAZE_B
/path_cdb/ESPRESSO_FLAT_A.fits FSPECTRUM_A
/path_cdb/ESPRESSO_FLAT_B.fits FSPECTRUM_B
/path_cdb/ESPRESSO_ORDERPROFILE_A.fits ORDER_PROFILE_A
/path_cdb/ESPRESSO_ORDERPROFILE_B.fits ORDER_PROFILE_B
/path_cdb/ESPRESSO_FP_SEARCHED_LINE_TABLE_A.fits FP_SEARCHED_LINE_TABLE_FP_FP_A
/path_cdb/ESPRESSO_FP_SEARCHED_LINE_TABLE_A.fits FP_SEARCHED_LINE_TABLE_FP_FP_A
Fibre contamination frames: those frames are characterized by DPR.TYPE='CONTAM'

[path_raw/ESPRE.2018-03-07T07:49:32.173.fits CONTAM_FP]
[path_cdb/ESPRESSO_SINGLEHR_1x1_TH_REF_LINE_TABLE_A.fits REF_LINE_TABLE_A]
[path_cdb/ESPRESSO_SINGLEHR_1x1_TH_REF_LINE_TABLE_B.fits REF_LINE_TABLE_B]
[path_cdb/ESPRESSO_S2D_BLAZE_FP_FP_A.fits S2D_BLAZE_FP_FP_A]
[path_cdb/ESPRESSO_S2D_BLAZE_FP_FP_B.fits S2D_BLAZE_FP_FP_B]

Cross fibre efficiency frames: those frames are characterized by DPR.TYPE='EFF,SKY,SKY'

[path_cdb/ESPRESSO_4x2_CCD_geom_config.fits CCD_GEOM]
[path_cdb/ESPRESSO_MULTIMR_4x2_inst_config.fits INST_CONFIG]

science frames: those frames are characterized by DPR.TYPE='OBJECT' or DPR.TYPE='SKY'

[path_raw/ESPRE.2018-02-28T09:12:23.678.fits OBJ_FP]
[path_cdb/ESPRESSO_4x2_CCD_geom_config.fits CCD_GEOM]
[path_cdb/ESPRESSO_MULTIMR_4x2_inst_config.fits INST_CONFIG]
[path_cdb/ESPRESSO_EXTINCTION_TABLE.fits EXT_TABLE]
[path_cdb/ESPRESSO_mask_lut.fits MASK_LUT]
[path_cdb/ESPRESSO_G2.fits MASK_TABLE]
[path_cdb/ESPRESSO_K5.fits MASK_TABLE]
[path_cdb/ESPRESSO_M2.fits MASK_TABLE]
[path_cdb/ESPRESSO_MASTER_BIAS_RES.fits MASTER_BIAS_RES]
[path_cdb/ESPRESSO_STD_TABLE.fits STD_TABLE]

We describe below a typical data reduction sequence using EsoRex. In this section we assume that the user sets in the EsoRex configuration file ($HOME/esorex/esorex.rc) the flag suppress-prefix to TRUE, so that the
pipeline product file names have standard names, with extension .fits for images and tables. We suggest to verify to have the flag `readonly` set to FALSE, if the user would like to run the same recipe several times with EsoRex having standard names for product files. This setting allows the pipeline to overwrite previously generated products. In the following we indicate only those frames involved in the data reduction cascade, suggesting the user to rename them according to their PRO.CATG, INS.MODE and DET.BINX, and to remove the other products after each recipe execution.

1. The user may start to generate a master bias. Raw dark frames may be put in an ASCII file, mbias_sof. This file will look like as follows:

   /path_raw/ESPRE.2018-09-02T11:00:27.443.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:01:34.278.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:02:41.057.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:03:49.938.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:04:58.840.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:06:05.651.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:07:14.526.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:08:21.291.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:09:28.075.fits BIAS
   /path_raw/ESPRE.2018-09-02T11:10:34.907.fits BIAS
   /path_cdb/ESPRESSO_2x1_CCD_geom_config.fits CCD_GEOM
   /path_cdb/ESPRESSO_SINGLEHR_2x1_inst_config_1slice.fits INST_CONFIG

   Then the user can generate the bias residuals frame with the command:

   `esorex espdr_rec_mbias mbias_sof`

   This command will generate two files: ESPRESSO_master_bias.fits (PRO.CATG=MASTER_BIAS), a master bias frame, and ESPRESSO_master_bias_res.fits (PRO.CATG=MASTER_BIAS_RES), a frame describing the master bias residuals. For convenience we indicate with /path_cdb the full path to a directory containing relevant data reduction products.

   `mv ESPRESSO_master_bias_res.fits /path_cdb`

2. Then the user can generate the master dark with the command:

   `esorex espdr_rec_mdark mdark_sof`

   This command will generate the file: ESPRESSO_hot_pixels.fits (PRO.CATG=HOT_PIXEL_MASK), a hot pixel mask, ESPRESSO_master_dark.fits (PRO.CATG=MASTER_DARK), a master dark. For convenience we indicate with /path_cdb the full path to a directory containing relevant data reduction products.

   `mv ESPRESSO_hot_pixels.fits /path_cdb`

   `rm -rf out*fits *.log`

3. Then the user can generate the bad pixel map frame with the command:

   `esorex espdr_rec_bpm bdmap_sof`

   This command will generate the file: ESPRESSO_bad_pixel_map.fits (PRO.CATG=BAD_PIXEL_MAP), a bad pixel map. For convenience we indicate with /path_cdb the full path to a directory containing relevant data reduction products.

   `mv ESPRESSO_bad_pixel_map.fits /path_cdb`

---

9 By default installation in the EsoRex configuration file ($HOME/esoex/esorex.rc) the flag `suppress-prefix` 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 a four digit number, and extention .fits for images and tables.
esorex espdr_led_ff ledff_sof
This command will generate the file: ESPRESSO_bad_pixels.fits (PRO.CATG=BAD_PIXEL_MASK).
mv ESPRESSO_bad_pixels.fits /path_cdb
rm -rf out*fits *.log

4. Then the user can find the order traces with the command:
esorex espdr_orderdef orderdef_sof
This command will generate the files: ESPRESSO_ORDER_TABLE_A/B.fits (PRO.CATG=ORDER_TABLE_A/B).
mv ESPRESSO_ORDER_TABLE_*.fit /path_cdb
rm -rf out*fits *.log

5. Then the user can process the flat frames with the command:
esorex espdr_mflat mflat_sof
This command will generate several files: ESPRESSO_ORDERPROFILE_A/B.fits (PRO.CATG=ESPRESSO_ORDERPROFILE_A/B), ESPRESSO_BLAZE_A/B.fits (PRO.CATG=ESPRESSO_BLAZE_A/B)
mv ESPRESSO_ORDERPROFILE_A/B.fits /path_cdb
mv ESPRESSO_BLAZE_A/B.fits /path_cdb
rm -rf out*fits *.log

6. Then the user can process the wave FP frames with the command:
esorex espdr_wave_FP wave_FP_sof
This command will generate several files: ESPRESSO_S2D_FP_FP_A/B.fits (PRO.CATG=S2D_FP_FP_A/B), ESPRESSO_S2D_BLAZE_FP_FP_A/B.fits (PRO.CATG=S2D_BLAZE_FP_FP_A/B), ESPRESSO_FP_SEARCHEDLINE_TABLE_FP_FP_A/B.fits (PRO.CATG=FP_SEARCHEDLINE_TABLE_FP_FP_A/B)
mv ESPRESSO_FP_SEARCHEDLINE_TABLE_FP_FP_A/B.fits /path_cdb
rm -rf out*fits *.log

7. Then the user can process the wave THAR frames with the command:
esorex espdr_wave_THAR wave_THAR_sof
This command will generate several files:

ESPERSSO_S2D_THAR_FP_A/B.fits
ESPERSSO_S2D_BLAZE_THAR_FP_A/B.fits
ESPERSSO_WAVE_MATRIX_A.fits
ESPERSSO_AIR_WAVE_MATRIX_A.fits
ESPERSSO_DLL_MATRIX_A.fits
ESPERSSO_AIR_DLL_MATRIX_A.fits
ESPERSSO_WAVE_TABLE_A.fits
ESPERSSO_FP_FITTEDLINE_TABLE_A.fits
ESPERSSO_THAR_LINE_TABLE_A.fits
ESPERSSO_LINE_TABLE_RAW_A.fits
mv ESPRESSO_AIR_WAVE_MATRIX_A.fits /path_cdb
rm -rf out*fits *.log

8. Then the user can process the contamination frames with the command:
   esorex espdr_contam contam_sof
   This command will create the frames: ESPRESSO_CONTAM_S2D_A/B.fits
   ESPRESSO_CONTAM_FP_B.fits
   mv ESPRESSO_CONTAM_FP_B.fits /path_cdb
   rm -rf out*fits *.log

9. Then the user can process the twilight sky frames to determine the fibre relative efficiency with the command:
   esorex espdr_eff_ab eff_ab_sof
   This command will create the frames: ESPRESSO_S2D_BLAZE_A/B.fits
   ESPRESSO_REL_EFF_B.fits
   mv ESPRESSO_REL_EFF_B.fits /path_cdb
   rm -rf out*fits *.log

10. Then the user can process the flux standard star frames to determine the fibre absolute efficiency with the command:
    esorex espdr_flux flux_sof
    This command will create the frames:
    mv ESPRESSO_REL_EFF_B.fits /path_cdb
    rm -rf out*fits *.log

11. Finally the user can process the science frames:
    esorex espdr_sci_red sci_red_sof This command will create the frames:
6 Known problems

So far we are aware of the following pipeline problems:

- The data reduction chain in SINGLE mode is quite slow it may take around one hour. We have measured the following execution times (in minutes) on the QC cluster.

<table>
<thead>
<tr>
<th>Recipe</th>
<th>1x1</th>
<th>1x2</th>
</tr>
</thead>
<tbody>
<tr>
<td>espdr_mbias</td>
<td>5.9</td>
<td>2.8</td>
</tr>
<tr>
<td>espdr_mdark</td>
<td>6.0</td>
<td>3.0</td>
</tr>
<tr>
<td>espdr_led_ff</td>
<td>15.6</td>
<td>8.3</td>
</tr>
<tr>
<td>espdr_orderdef</td>
<td>4.2</td>
<td>2.0</td>
</tr>
<tr>
<td>espdr_mflat</td>
<td>16.0</td>
<td>6.6</td>
</tr>
<tr>
<td>espdr_cal_contam</td>
<td>2.4</td>
<td>1.2</td>
</tr>
<tr>
<td>espdr_wave_FP</td>
<td>1.7</td>
<td>0.9</td>
</tr>
<tr>
<td>espdr_wave_THAR</td>
<td>1.5</td>
<td>0.8</td>
</tr>
<tr>
<td>espdr_wave_LFC</td>
<td>1.6</td>
<td>0.7</td>
</tr>
<tr>
<td>espdr_cal_eff_ab</td>
<td>2.2</td>
<td>1.1</td>
</tr>
<tr>
<td>espdr_cal_flux</td>
<td>2.6</td>
<td>1.4</td>
</tr>
<tr>
<td>espdr_sci_red</td>
<td>3.2</td>
<td>1.9</td>
</tr>
</tbody>
</table>

Future releases will tackle this problem, where appropriate, using OpenMP. This mean that in the future user may have faster reduction on multi-core platforms.

- Observations of flux STD star not contained in the reference catalog will make the recipe `espdr_cal_flux` fail. We found this problem seldom during commissioning. We expect it will not occur during operations.

- Seldom the reduction of THAR,FP data may fail. We found this seldom during commissioning. We expect this should not occur during operations.

For future releases we plan to implement the following improvements:

- Speed up data reduction.
- Implement alternative sky subtraction method in which the sky spectrum is fitted by a low-order polynomial (to minimize readout noise on the sky-subtracted science spectrum); this is intended for non-RV applications.
- Improve robustness of the `espdr_wave_THAR` recipe by adding a few more Th lines in spectral orders that have only a few valid lines
- Fine-tune CCF residual barycenter computation.
- Build and test new cross-correlation masks for K and M stars based on actual ESPRESSO spectra.
- Skip CCF computation in science recipe if user-provided redshift > 0 (extragalactic object).
- Generate all possible pipeline products for `espdr_cal_flux` even if the observed standard star is not in the reference catalog.
7 Instrument Data Description

The ESPRESSO instrument produces raw data in five different configurations or modes. Those are defined by the main instrument mode and detector binning. The corresponding FITS keywords are:

- HIERARCH ESO INS MODE: instrument mode, defining the fiber head used and the spectral resolution
- HIERARCH ESO DET BINX: detector binning in cross-dispersion direction
- HIERARCH ESO DET BINY: detector binning in main dispersion direction

The five available configurations are:

- SINGLEHR mode and 1x1 binning (HR11)
- SINGLEHR mode and 2x1 binning (HR21)
- SINGLEUHR mode and 1x1 binning (UHR)
- MULTIMR mode and 4x2 binning (MR42)
- MULTIMR mode and 8x4 binning (MR84)

The ESPRESSO pipeline considers these five configurations as independent instruments for the purpose of data reduction. In practice this means that a science frame in a given configuration requires a complete calibration dataset in the same configuration to be processed by the pipeline. As part of the ESPRESSO operational scheme at the observatory, the acquisition of the required calibration frames is automatically triggered based on the science data obtained during the night.

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

- A set of 10 BIAS frames, acquired within 48 hours of the science observation
- A set of 5 DARK frames, acquired within 3 months of the science observation
- A set of 2x5 LED frames (2 different exposure times), acquired within 3 months of the science observation
- 2 order definition frames (one per fiber), acquired within 48 hours of the science observation
- A set of 2x10 FLAT frames (one set per fiber), acquired within 48 hours of the science observation
- A Fabry-Perot (FP) wavelength calibration frame, acquired within 48 hours of the science observation
- A Thorium-FP wavelength calibration frame, acquired within 48 hours of the science observation
- A FP-Thorium wavelength calibration frame, acquired within 48 hours of the science observation
- A fiber-to-fiber relative efficiency exposure, acquired within 3 months of the science observation
• A spectrophotometric standard star exposure, acquired within 3 months of the science observation

Optionally, the following calibration frames are needed for wavelength calibration with the laser frequency comb (LFC):

• An LFC-FP wavelength calibration frame, acquired within 48 hours of the science observation
• An FP-LFC wavelength calibration frame, acquired within 48 hours of the science observation

Optionally, the following calibration frame is needed to correct for cross-fiber contamination by the FP in simultaneous-reference mode (unnecessary for now according to commissioning results):

• An FP contamination frame, acquired within 48 hours of the science observation

It is also necessary to have a set of static calibration data to hand (see next Section).

The following sections provides a brief description of each raw data type involved in the data reduction chain.

7.1 BIAS frames

Bias frames are zero exposure frames taken to measure the readout noise, the mean bias level, and fixed-pattern structure in the bias level. In ESPRESSO the mean bias level and readout noise in any raw frame are best obtained from the overscan regions. The master bias is thus not used as such in the reduction cascade. A master bias residual frame is used instead to subtract the fixed residual bias level structure across detector outputs. This latter product is generated by the bias recipe. The residual bias level pattern is not negligible for low-SNR science exposures.

7.2 DARK frames

Dark frames are taken to measure the detector dark current and identify hot pixels. The mean dark current level per output and the hot pixels are stored in a hot pixel map that is used by the other pipeline recipes along the reduction chain. The exposure time is long.

7.3 Detector Flat-field (LED) frames

Detector flat-field (illumination with LEDs) are acquired to characterize CCD cosmetics (bad pixel mask) and measure the conversion factor. Several (at least five) frames and (at least) two different exposure times are required.
7.4 ORDERDEF frames

The following frames are acquired:

- **ORDERDEF_A**: Order/slice definition frame for fiber A (identification and tracing). A continuum light source is used to illuminate fiber A, while fiber B is dark (separate frames for different fibers make the automatic identification of orders and slices easier).

- **ORDERDEF_B**: Order/slice definition frame for fiber B (identification and tracing). A continuum light source is used to illuminate fiber B, while fiber A is dark (separate frames for different fibers make the automatic identification of orders and slices easier).

7.5 FLAT frames

The following frames are acquired:

- **FLAT_A**: Flat-field, blaze and order profile frame for fiber A. A continuum light source is used to illuminate fiber A, while fiber B is dark. Several exposures are required to reach sufficiently high SNR. The spectrum is used to derive: 1) the order profile in cross-dispersion direction, 2) the spectral flat-field, and 3) the blaze function of the spectrograph.

- **FLAT_B**: Flat-field, blaze and order profile frame for fiber B. A continuum light source is used to illuminate fiber B, while fiber A is dark. Several exposures are required to reach sufficiently high SNR. The spectrum is used to derive: 1) the order profile in cross-dispersion direction, 2) the spectral flat-field, and 3) the blaze function of the spectrograph.

7.6 FP-FP frames

**FP_FP**: These frames are obtained illuminating both fibers with a Fabry Perot.

7.7 THAR-FP frames

**THAR_FP**: These frame is obtained illuminating fiber A with a ThAr lamp and fiber B with a Fabry Perot.

7.8 FP-THAR frames

**THAR_FP**: These frame is obtained illuminating fiber B with a ThAr lamp and fiber A with a Fabry Perot.

7.9 LFC-FP frames

**LFC_FP**: These frame is obtained illuminating fiber A with a laser comb and fiber A with a Fabry Perot.
7.10 FP-LFC frames

FP_LFC: These frames are obtained illuminating fiber B with a laser comb and fiber A with a Fabry Perot.

7.11 Contamination by simultaneous reference frames

The following frames are acquired:

- CONTAM_THAR: Measurement of contamination light induced on fiber A by ThAr simultaneous reference on fiber B. Fiber A is dark, while ThAr light is injected into fiber B as in a science exposure with simultaneous reference (i.e. same flux level).
- CONTAM_FP: Similar to previous one but fiber B is illuminated by a Fabry Perot.
- CONTAM_LFC: Similar to CONTAM_THAR but fiber B is illuminated by a laser comb.

7.12 Fiber-to-fiber Relative Efficiency frames

EFF_AB: Relative efficiency of fiber B vs. fiber A as a function of wavelength. The telescope is pointed at daylight or twilight sky and skylight is injected into both fibers. The obtained relative efficiency is used to subtract the sky on science exposures with simultaneous sky.

7.13 Spectrophotometric Calibration (FLUX) frames

FLUX_CALIB: A spectrophotometric standard star is observed as in a science exposure with sky, and the absolute efficiency of the instrument as a function of wavelength is computed from the comparison between observed spectrum and reference flux table.

7.14 SCIENCE frames

The following frames are acquired:

- SCIENCE_FP: Science exposure (target on fiber A) with Fabry-Perot simultaneous reference on fiber B.
- SCIENCE_SKY: Science exposure (target on fiber A) with simultaneous sky on fiber B.
8 Static Calibration Data

In the following section ancillary data required for ESPRESSO data reduction are listed. For each of them we indicate the corresponding value of the HIERARCH ESO PRO CATG, in short PRO.CATG, FITS keyword. This has to be used to identify the frames listed in the Set of Frames (see Section 5.2.3, page 26).

8.1 CCD Geometry Table

These are the static CCD configuration tables describing the CCD geometry. The table contains the number and sizes of the detectors, outputs and prescan and overscan regions. There is a table for each of the supported instrument modes. Its PRO.CATG is CCD_GEOM.

8.2 Instrument Configuration Table

These are the static instrument configuration tables providing the DRL recipes with all necessary input parameters that are intimately linked to the instrument configuration being used. There will be one such table per instrument configuration (i.e. five for ESPRESSO). Its PRO.CATG is INST_CONFIG.

8.3 Led Flat Field gain windows Table

These static tables indicate for each detector read-out region the windows where the conversion factor is computed. Their PRO.CATG is LED_FF_GAIN_WINDOWS.

8.4 Wavelength Matrix Images

These are the wavelength calibration arrays (one per fiber) in S2D format, with the wavelength of each extracted pixel stored as data value. These are used as static input frames in the flat and wavelength (THAR/FP or FP/THAR) data reduction. Their PRO.CATG is STATIC_WAVE_MATRIX_A/B.

8.5 DLL Matrix Images

These are the pixels width in wavelength calibration arrays (one per fiber) in S2D format, with the width in wavelength of each extracted pixel stored as data value. Their PRO.CATG is STATIC_DLL_MATRIX_A/B.

8.6 THAR line tables

These are the static ThAr line tables (one per fiber) containing the approximate positions, the identifications, wavelengths and grouping flags of the emission lines found in the ThAr extracted spectra (S2D). The table is not expected to change over the lifetime of the instrument owing to the high long-term stability of ESPRESSO. Their PRO.CATG is REF_LINE_TABLE_A/B.
8.7 Flux Standard Star tables

This is the table containing absolute spectral energy distributions of a sample of spectrophotometric standard stars. Its PRO.CATG is STD_TABLE.

8.8 Extinction Table

This is the table containing the atmospheric extinction curve for Paranal, as provided in Patat et al. (2011), A&A 527, 91 (their Appendix B). Its PRO.CATG is EXT_TABLE.

8.9 Flux Template Tables

This is the table containing observed spectral energy distributions of a sample of reference stars with different spectral types spanning late-F to early-M. Suitable stars are solar-metallicity dwarf stars of spectral types F to M, observed at high SNR and at low airmass. To build the flux template, the S2D flux of the star is summed in each spectral order and normalized to one at an arbitrary wavelength (e.g. 550 nm). The flux template is then simply the integrated normalized flux in each spectral order. Its PRO.CATG is FLUX_TEMPLATE.

8.10 CCF Template Tables

These are the line masks used by the cross-correlation process (one file per mask). They consist of a list of laboratory wavelengths for the spectral lines of interest, along with their relative depth (contrast) with respect to the continuum. Depths are used as weights in the CCF process since Doppler precision is proportional to line depth. Note that the properties of stellar spectra change relatively slowly with spectral type, so that the use of one mask per main spectral type (G, K, M) does not significantly degrade performances. Their PRO.CATG are MASK_TABLE.
9 Data Reduction

We give below an overview of the global reduction cascade, starting from basic calibrations up to science reduction. The ESPRESSO association map is shown on Figure 9.0.0.

- Detector bias level and readout noise are measured on stacked BIAS frames (master bias). The overscan zone is checked to see whether it can be used to reliably determine bias level. If so, the standard way to subtract bias on all kinds of frames will be to use the overscan. In addition the BIAS residuals obtained by subtracting the overscan form the master bias, are used for bias correction in different types of frames.

- DARK frames are used to measure the average dark current and obtain a list of hot pixels. A master dark is generated via stacking and sigma-clipping to remove the cosmics. The dark current and hot pixels are computed on the master frame.

- A bad pixel mask is obtained from the analysis of LED_FF frames taken with two different exposure times (linearity check). The gain is also measured from the relation between flux level and standard deviation of pixel values. For each exposure time, a set of minimum 5 LED_FF frames is taken to remove the cosmics. Then the masters are used for further computation.

- ORDERDEF frames (one per fibre) are used to identify and trace spectral orders and slices on the detector. Only the order/slice center is relevant at this stage. They are fitted with a low degree polynomial, which coefficients are saved in the recipe products.

- For each fibre, the order profile in cross-dispersion direction is found using high-SNR, co-added FLAT frames. Then, the orders are extracted from the FLAT frames using this profile, and the spectral flat-field is generated. The (extracted) blaze function is obtained through smoothing of the FLAT spectra and correction for the spectral energy distribution of the calibration lamp. The foreseen strategy for order extraction assumes that 1) main-dispersion direction runs approximately parallel to CCD rows/columns, and 2) slit image tilt is close to zero with respect to CCD columns/rows. In this case, order extraction becomes extremely simple and does not require wavelength calibration frames to track different positions along the slit. The ESPRESSO optical design makes this strategy possible (line tilt very close to zero), and even recommended owing to its simplicity. This method has been successfully applied to HARPS and other radial-velocity spectrographs.

- CONTAM frames are used to measure cross-fibre contamination on fibre A from the simultaneous reference on fibre B (ThAr lamp, laser comb or Fabry-Perot). Contamination frames are used during extraction in the science reduction.

- The relative efficiency of channels A and B as a function of wavelength is measured using EFF_AB frames, which are obtained through blue sky observations. This calibration product is used during sky subtraction in the science reduction.

- The wavelength calibration for both fibres is determined using WAVE frames. It is done in two steps: first with the Fabry-Perot on both fibres and then with one of the fibres illuminated by ThAr lamp. In addition the wavelength solution can be computed using frames with Laser Frequency Comb, for which a previous ThAr wavelength solution is needed. On fibre A (science fibre), the wavelength solution is particularly important since it is used to calibrate the science spectrum and therefore sets the radial velocity zero point.
On fibre B, the spectrum mainly serves as drift reference spectrum for the drift measurement on science frames with simultaneous reference.

- FLUX_CALIB frames are used to compute the absolute efficiency of the instrument using spectrophotometric standard stars. The efficiency curve is used in the science reduction to calibrate the science spectrum in flux. ESPRESSO being a fibre-fed instrument with a fixed sky aperture of 1.0 arcsec or less, the precision of the flux calibration is expected to be low because of highly variable slit losses.

- Finally, SCIENCE frames are of two different sub-types depending on the source of light on the fibre B: sky or Fabry-Perot. Science reduction makes use of all calibration products listed above and generates extracted S2D spectra and merged, rebinned S1D spectra, together with S2D and S1D error and quality maps. Finally, the cross-correlation function (CCF) of the S2D spectrum is computed and the radial velocity is obtained from a Gaussian fit to the CCF.
Figure 9.0.0: The cascade of the ESPRESSO pipeline recipes. For the calibration recipes, only the products used later in the reduction chain are listed. The science recipes have all the products listed.
10 Pipeline Recipes Interfaces

In this section we provide for each recipe examples of the required input data (and their tags). In the following we assume that /path_file_raw/filename_raw.fits and /path_file_cdb/filename_cdb.fits are existing FITS files (e.g. /data1/espresso/com2/ESPRE.2018-02-01T02:54:04.353.fits and /cal/esdr/cal/MASTER_BIAS_RES.fits).

We also provide a list of the pipeline products for each recipe, indicating their default recipe name (eventually set by esorex to a given standard), the value of the FITS keyword HIERARCH ESO PRO CATG (in short PRO.CATG) and a short description.

The relevant keywords are PRO.CATG, used to classify each frame, and to associate to each raw frame the proper calibration frame:

<table>
<thead>
<tr>
<th>Association keyword</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIERARCH ESO INS MODE</td>
<td>mode</td>
</tr>
<tr>
<td>HIERARCH ESO DET BINX</td>
<td>detector bin X</td>
</tr>
<tr>
<td>HIERARCH ESO EXPTIME</td>
<td>Integration time</td>
</tr>
<tr>
<td>HIERARCH ESO OBS TARG NAME</td>
<td>Indicates the observed object ID</td>
</tr>
</tbody>
</table>

For each recipe we also list in a table the input parameters (as they appear in the recipe configuration file), the corresponding aliases (the corresponding names to be eventually set on command line) and their default values. Also quality control parameters are listed. Those are stored in relevant pipeline products. More information on instrument quality control can be found on www.eso.org/qc.

10.1 espdr_mbias

10.1.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>BIAS</td>
<td>5...n</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>

10.1.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>MASTER_BIAS</td>
<td>qc</td>
<td>Master bias</td>
</tr>
<tr>
<td>1</td>
<td>MASTER_BIAS_RES</td>
<td>cdb</td>
<td>Master bias residuals</td>
</tr>
</tbody>
</table>

10.1.3 Quality control

This recipe computes the following QC parameters (on the image extension n, and read-out region i,j): QC EXTn ROXi ROYj BIAS RON, read out noise [ADU]
QC RES TEST, overall quality check on residual computation
QC EXTn ROXi ROYj BIAS RON EL, read out noise [e-]
QC EXTn ROXi ROYj BIAS MEAN EL, mean value [e-]
QC EXTn ROXi ROYj RES MEAN EL, mean value of residuals [e-]
QC EXTn ROXi ROYj RES STDEV EL, rms of residuals [e-]
QC EXTn ROXi ROYj OVS C RON EL, read out noise on overscan region [e-]
QC EXTn ROXi ROYj RES MEAN, mean value of residuals [ADU]
QC EXTn ROXi ROYj RES STDEV, rms of residuals [ADU]
QC EXTn ROXi ROYj OVS C RON, read out noise on overscan region [ADU]
QC EXTn ROXi ROYj BIAS MEAN, mean bias [ADU]
QC EXTn ROXi ROYj BIAS STRUCTX, X structure
QC EXTn ROXi ROYj BIAS STRUCTY, Y structure
QC BIAS OUTLIERS, number out utliers on bias
QC OVS C OUTLIERS, number of outliers on region
QC BIAS RON CHECK, quality check on RON computation
QC BIAS MEAN CHECK, quality check on bias mean computation
QC EXTn ROXi ROYj RES TEST, quality check on residual computation on read-out region n
QC BIAS METHOD, data reduction method used
QC BIAS CHECK, overall quality check assessment.

10.1.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_mbias

10.2 espdr_mdark

10.2.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>DARK</td>
<td>5...n</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>

10.2.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PROC.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HOT_PIXEL_MASK</td>
<td>cdb</td>
<td>Hot pixels mask</td>
</tr>
<tr>
<td>1</td>
<td>MASTER_DARK</td>
<td>qc</td>
<td>Master dark</td>
</tr>
</tbody>
</table>
10.2.3 Quality control

This recipe computes the following QC parameters (on the image extension n, and read-out region i,j): QC EXTn ROXi ROYj DARK MEAN, Mean Dark [ADU/px/hour]
QC EXTn ROXi ROYj HOTPIX NB, number of hot pixels
QC EXTn COSMIC RATE PIX, number of cosmic hits per pixel per hour
QC EXTn COSMIC RATE CM2, number of cosmic hits per cm2 per hour
QC DARK COSMIC CHECK, check on cosmic rate computation results
QC DARK MEAN CHECK, check on mean dark results
QC DARK HOTPIX CHECK, check on hotpixel detection
QC DARK CHECK, overall check on dark

10.2.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_mdark

10.3 espdr_led_ff

10.3.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>LED_FF</td>
<td>5...n</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>LED_FF_GAIN_WINDOWS</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>HOT_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>

10.3.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BAD_PIXEL_MASK</td>
<td>cdb</td>
<td>Bad pixels mask</td>
</tr>
</tbody>
</table>

10.3.3 Quality control

This recipe computes the following QC parameters (on the image extension n, and read-out region i,j):
QC EXTn ROXi ROYj MAX FLUX, max flux on raw image [ADU]
QC EXTn ROXi ROYj CONAD, conversion factor [e-/ADU]
QC EXTn ROXi ROYj BADPIX NB, bad pixel number
QC EXTn ROXi ROYj MASTER MAX FLUX, max flux on master [ADU]
QC EXTn ROXi ROYj MASTER MIN FLUX, min flux on master [ADU]
QC EXTn ROXi ROYj MASTER MEAN FLUX, mean flux on master [ADU]
QC MASTER MIN FLUX, overall min flux on master [ADU]
QC MASTER MAX FLUX, overall max flux on master [ADU]
QC SATURATION CHECK, check on frame saturation
QC LEDFF MIN FLUX CHECK, check on minimum flux
QC LEDFF BADPIX CHECK, check on bad pixels
QC LEDFF CONAD CHECK, check on conad
QC LEDFF CHECK, overall check on results.

10.3.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_led_ff

10.4 espdr_orderdef

10.4.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>ORDERDEF_A</td>
<td>1</td>
<td>any</td>
</tr>
<tr>
<td>raw</td>
<td>ORDERDEF_B</td>
<td>1</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>HOT_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BAD_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>

10.4.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ORDER_TABLE_A</td>
<td>cdb</td>
<td>Order table tracing fibre A</td>
</tr>
<tr>
<td>1</td>
<td>ORDER_TABLE_B</td>
<td>cdb</td>
<td>Order table tracing fibre B</td>
</tr>
</tbody>
</table>

10.4.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

QC ORDERm EXTn POS, order position [pix]
QC ORDERm EXTn RES MIN, min residuals [pix]
QC ORDERm EXTn RES MAX, max residuals [pix]
QC ORDERm EXTn RES STDEV, rms residuals [pix]
QC ORDERm EXTn NB, physical order number
QC ORDER NB, Number of orders in the image
QC EXTn ROXi ROYj MAX FLUX, Max flux, raw image [ADU]
QC SATURATION CHECK, check on saturation
QC ORDERDEF ORDER CHECK, check on number of detected orders
QC ORDERDEF STDEV CHECK, check on rms residuals
QC ORDERDEF MIN CHECK, check on min residuals
QC ORDERDEF MAX CHECK, check on max residuals
QC ORDERDEF CHECK, overall check

10.4.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_orderdef

10.5 espdr_mflat

10.5.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>FLAT_A</td>
<td>5..n</td>
<td>any</td>
</tr>
<tr>
<td>raw</td>
<td>FLAT_B</td>
<td>5..n</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STATIC_WAVE_MATRIX_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STATIC_WAVE_MATRIX_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>HOT_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BAD_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_B</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>

10.5.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FF_BACKGROUND_MAP_A</td>
<td>cdb</td>
<td>Background map for fibre A</td>
</tr>
<tr>
<td>1</td>
<td>FS2D_A</td>
<td>cdb</td>
<td>Order by order extracted spectrum for fibre A</td>
</tr>
<tr>
<td>2</td>
<td>BLAZE_A</td>
<td>cdb</td>
<td>Blaze for fibre A</td>
</tr>
<tr>
<td>3</td>
<td>FSPECTRUM_A</td>
<td>cdb</td>
<td>Order by order extracted spectrum for fibre A</td>
</tr>
<tr>
<td>4</td>
<td>ORDER_PROFILE_A</td>
<td>cdb</td>
<td>Order profile for fibre A</td>
</tr>
<tr>
<td>5</td>
<td>FF_BACKGROUND_MAP_B</td>
<td>cdb</td>
<td>Background map for fibre B</td>
</tr>
<tr>
<td>6</td>
<td>FS2D_B</td>
<td>cdb</td>
<td>Order by order extracted spectrum for fibre B</td>
</tr>
<tr>
<td>7</td>
<td>BLAZE_B</td>
<td>cdb</td>
<td>Blaze for fibre B</td>
</tr>
<tr>
<td>8</td>
<td>FSPECTRUM_B</td>
<td>cdb</td>
<td>Order by order extracted spectrum for fibre B</td>
</tr>
<tr>
<td>9</td>
<td>ORDER_PROFILE_B</td>
<td>cdb</td>
<td>Order profile for fibre B</td>
</tr>
</tbody>
</table>
10.5.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

- QC ORDERm FLAT RMS, FLAT RMS in the order
- QC ORDERm SNR, SNR in the order
- QC ORDERm COSMIC NB, Cosmics number in the order
- QC EXTn ROXi ROYj MAX FLUX, Max flux, raw image [ADU]
- QC EXTn BKGR MEAN, Bkgr mean in the extension [e-]
- QC EXTn BKGR MIN, Bkgr min in the extension [e-]
- QC SATURATION CHECK, quality check on saturation
- QC FLAT RMS CHECK, quality check on rms
- QC FLAT SNR CHECK, quality check on SNR
- QC FLAT BKGR CHECK, background quality check
- QC FLAT CHECK, overall quality check

10.5.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_mflat

10.6 espdr_wave_FP

10.6.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>FP_FP</td>
<td>1</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>HOT_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BAD_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_PROFILE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_PROFILE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>FSPECTRUM_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>FSPECTRUM_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BLAZE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BLAZE_B</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>

10.6.2 Output
10.6.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

- QC ORDERm LINES NB, Number of valid lines in order m
- QC ORDERm COSMIC NB, Number of cosmics detected in order m
- QC EXTn ROXi ROYj MAX FLUX, Max flux in the raw image [ADU]
- QC SATURATION CHECK, quality check on saturation
- QC LINES NB, Total number of valid lines
- QC COSMIC NB, number of cosmics
- QC LINES TOT CHECK, check on total number of valid lines
- QC WAVE CHECK, overall quality check

10.6.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_wave_FP

10.7 espdr_wave_THAR

10.7.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>THAR_FP</td>
<td>1</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STATIC_DLL_MATRIX_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STATIC_DLL_MATRIX_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STATIC_WAVE_MATRIX_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STATIC_WAVE_MATRIX_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>REF_LINE_TABLE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>REF_LINE_TABLE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>HOT_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BAD_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_A</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>
And for the other fibre a similar input where THAR_FP is replaced by FP_THAR, and FP_SEARCHED_LINE_TABLE_FP_FP_A is replaced by FP_SEARCHED_LINE_TABLE_FP_FP_B.

### 10.7.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>S2D_THAR_FP_A</td>
<td>cdb</td>
<td>S2D Extracted spectrum for fibre A</td>
</tr>
<tr>
<td>1</td>
<td>S2D_THAR_FP_B</td>
<td>cdb</td>
<td>S2D Extracted spectrum for fibre B</td>
</tr>
<tr>
<td>2</td>
<td>S2D_BLAZE_THAR_FP_A</td>
<td>cdb</td>
<td>S2D Extracted blaze for fibre A</td>
</tr>
<tr>
<td>3</td>
<td>S2D_BLAZE_THAR_FP_B</td>
<td>cdb</td>
<td>S2D Extracted blaze for fibre B</td>
</tr>
<tr>
<td>4</td>
<td>WAVE_MATRIX_THAR_FP_A</td>
<td>cdb</td>
<td>Wave map (vacuum) for fibre A</td>
</tr>
<tr>
<td>5</td>
<td>AIR_WAVE_MATRIX_THAR_FP_A</td>
<td>cdb</td>
<td>Wave map (air) for fibre A</td>
</tr>
<tr>
<td>6</td>
<td>DLL_MATRIX_THAR_FP_A</td>
<td>cdb</td>
<td>DLL map (vacuum) fibre A</td>
</tr>
<tr>
<td>7</td>
<td>AIR_DLL_MATRIX_THAR_FP_A</td>
<td>cdb</td>
<td>DLL map (air) fibre A</td>
</tr>
<tr>
<td>8</td>
<td>WAVE_TABLE_THAR_FP_A</td>
<td>cdb</td>
<td>Table of detected lines for fibre A</td>
</tr>
<tr>
<td>9</td>
<td>FP_FITTED_LINE_TABLE_THAR_FP_A</td>
<td>cdb</td>
<td>Table of fit lines for fibre A</td>
</tr>
<tr>
<td>10</td>
<td>THAR_LINE_TABLE_THAR_FP_A</td>
<td>cdb</td>
<td>Thar line table for fibre A</td>
</tr>
<tr>
<td>11</td>
<td>LINE_TABLE_RAW_THAR_FP_A</td>
<td>cdb</td>
<td>Thar line table for fibre A</td>
</tr>
</tbody>
</table>

And for the other fibre producs are the same except that from Id 4 they refer to fibre B instead than A.

### 10.7.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

QC DRIFT DETn SLOPE_O  
QC DRIFT DETn SLOPE_O_ERR  
QC DRIFT DETn SLOPE_X  
QC DRIFT DETn SLOPE_X_ERR  
QC DRIFT DETn CHISQ  
QC DRIFT DETn REJECTED  
QC DRIFT DETn MEAN
QC DRIFT DETn MEAN_ERR
QC DRIFT DETn FLUX_RATIO
QC DRIFT DETn FLUX_RATIO_ERR
QC DRIFT DETn FIRST_ORDER
QC DRIFT DETn LAST_ORDER
QC WAVE DRIFT CHI2 CHECK, quality check on wave drift
QC ORDERm LINES NB, Number of valid lines in order m
QC ORDERm CHI2, Reduced CHI2 of ll solution for order m
QC ORDERm RMS, Dispertion of residuals around mean
QC ORDERS RESOL MEDIAN, Thar lines resolution median
QC LINES NB, Total number of valid lines
QC LINES ORDER MIN CHECK, quality check on min order residuals
QC LINES TOT CHECK, quality check on total lines number
QC THAR CHI2 CHECK, quality check on chi2
QC THAR RMS CHECK, quality check on rms THAR
QC WAVE RESOL CHECK, quality check on resolution
QC WAVE CHECK, overall quality check

10.7.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_wave_THAR

10.8 espdr_wave_LFC

10.8.1 Input

10.8.2 Output

10.8.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

10.8.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_wave_LFC

10.9 espdr_contam

10.9.1 Input
10.9.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CONTAM_S2D_A</td>
<td>qc</td>
<td>S2D Extracted spectrum for fibre A</td>
</tr>
<tr>
<td>1</td>
<td>CONTAM_S2D_B</td>
<td>qc</td>
<td>S2D Extracted spectrum for fibre B</td>
</tr>
<tr>
<td>2</td>
<td>CONTAM_FP</td>
<td>cdb</td>
<td>Contamination of fibre B on fibre A</td>
</tr>
</tbody>
</table>

10.9.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

QC ORDERm MAX FLUX, S2D flux max, fibre A
QC EXTn ROXi ROYj MAX FLUX, Max flux, raw image
QC SATURATION CHECK, quality check on saturation
QC CONTAM FLUX CHECK, quality check on flux
QC CONTAM CHECK, overall quality check

10.9.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_cal_contam

10.10 espdr_eff_ab

10.10.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>EFF_AB</td>
<td>1</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>
10.10.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>S2D_BLAZE_EFF_A</td>
<td>qc</td>
<td>S2D Extracted blaze spectrum for fibre A</td>
</tr>
<tr>
<td>1</td>
<td>S2D_BLAZE_EFF_B</td>
<td>qc</td>
<td>S2D Extracted blaze spectrum for fibre B</td>
</tr>
<tr>
<td>2</td>
<td>REL_EFF_B</td>
<td>cdb</td>
<td>fibre B relative to fibre A</td>
</tr>
</tbody>
</table>

10.10.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

- QC ORDERm SNR, SNR in the order m
- QC ORDERm COSMIC NB, Cosmics number in the order
- QC EXTn ROXi ROYj MAX FLUX, Max flux, raw image [ADU]
- QC EXTn BKGR MEAN, Bkgr mean in the extension [e-]
- QC EXTn BKGR MIN, Bkgr min in the extension [e-]
- QC EXTn BKGR MAX, Bkgr max in the extension [e-]
- QC REL EFF MIN, Min relative efficiency
- QC REL EFF MAX, max relative efficiency
- QC REL EFF MIN CHECK, check on computation min efficiency
- QC REL EFF MAX CHECK, check on computation max efficiency
- QC EFFAB BKGR CHECK, check on background level
- QC SATURATION CHECK, check on saturation
- QC EFFAB SNR CHECK, check on SNR
- QC EFFAB CHECK, overall check on effab products

10.10.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_cal_eff_ab
10.11 espdr_cal_flux

10.11.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>FLUX</td>
<td>1</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>EXT_TABLE</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STD_TABLE</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>HOT_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BAD_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDERPROFILE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDERPROFILE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>FSPECTRUM_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>FSPECTRUM_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BLAZE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BLAZE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>WAVE_MATRIX_THAR_FP_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>WAVE_MATRIX_FP_THAR_B</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>

10.11.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>S2D_STD_A</td>
<td>qc</td>
<td>S2D Extracted spectrum for fibre A</td>
</tr>
<tr>
<td>1</td>
<td>S1D_STD_A</td>
<td>qc</td>
<td>S1D Extracted spectrum for fibre A</td>
</tr>
<tr>
<td>2</td>
<td>S1D_ENERGY_STD_A</td>
<td>cdb</td>
<td>S1D energy for fibre A</td>
</tr>
<tr>
<td>3</td>
<td>S2D_BLAZE_STD_A</td>
<td>cdb</td>
<td>S2D blaze spectrum for fibre A</td>
</tr>
<tr>
<td>4</td>
<td>AVG_FLUX_STD_A</td>
<td>cdb</td>
<td>average flux for fibre A</td>
</tr>
<tr>
<td>5</td>
<td>ABS_EFF_RAW_A</td>
<td>cdb</td>
<td>absolute raw efficiency for fibre A</td>
</tr>
<tr>
<td>6</td>
<td>ABS_EFF_A</td>
<td>cdb</td>
<td>absolute efficiency for fibre A</td>
</tr>
</tbody>
</table>

10.11.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

- QC ORDERm SNR, SNR in the order m
- QC ORDERm COSMIC NB, Cosmics number in the order m
- QC EXTn ROXi ROYj MAX FLUX, Max flux, raw image [ADU]
- QC EXTn BKGR MEAN, Bkgr mean in the extension [e-]
- QC EXTn BKGR MIN, Bkgr min in the extension [e-]
QC EXTn BKGR MAX, Bkg max in the extension [e-]
QC MEAN RAW ABS EFF, Mean raw absolute efficiency
QC MIN RAW ABS EFF, Min raw absolute efficiency
QC MAX RAW ABS EFF, Max raw absolute efficiency
QC MIN ABS EFF, Minimum absolute efficiency
QC MAX ABS EFF, Maximum absolute efficiency
QC MEAN ABS EFF, Mean absolute efficiency
QC MIN ABS EFF CHECK, Minimum absolute efficiency check
QC MAX ABS EFF CHECK, Maximum absolute efficiency check
QC SATURATION CHECK, Saturation [ADU] QC
QC BERV, Barycentric correction [km/s]
QC BJD, Barycentric Julian date (UTC) [JD]
QC BERVMAX, Barycentric max [km/s]
QC MIN ABS EFF CHECK
QC CALFLUX SNR CHECK, check on SNR computed value
QC MAX ABS EFF CHECK, check on max abs eff computed value
QC CALFLUX SEEING KW CHECK, check on seeing
QC CALFLUX CHECK, overall check on recipe

10.11.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_cal_flux

10.12 espdr_sci_red

10.12.1 Input

<table>
<thead>
<tr>
<th>type</th>
<th>TAG</th>
<th>n</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>OBJ_SKY/OBJ_THAR/OBJ_FP/OBJ_WLC</td>
<td>1</td>
<td>any</td>
</tr>
<tr>
<td>ref</td>
<td>CCD_GEOM</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>INST_CONFIG</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>MASK_TABLE</td>
<td>7</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>EXT_TABLE</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>MASK_LUT</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>STD_TABLE</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>ref</td>
<td>FLUX_TEMPLATE</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>MASTER_BIAS_RES</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>HOT_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>BAD_PIXEL_MASK</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_TABLE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_PROFILE_A</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>ORDER_PROFILE_B</td>
<td>1</td>
<td>match</td>
</tr>
<tr>
<td>cdb</td>
<td>FSPECTRUM_A</td>
<td>1</td>
<td>match</td>
</tr>
</tbody>
</table>
10.12.2 Output

<table>
<thead>
<tr>
<th>ID</th>
<th>PRO.CATG</th>
<th>type</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CCF_RESIDUALS_A</td>
<td>qc</td>
<td>residuals from CCF computation</td>
</tr>
<tr>
<td>1</td>
<td>DRIFT_MATRIX_B</td>
<td>qc</td>
<td>drift matrix relative to fibre B</td>
</tr>
<tr>
<td>2</td>
<td>S2D_BLAZE_A</td>
<td>cdb</td>
<td>S2D spectrum blazwe fibre A</td>
</tr>
<tr>
<td>3</td>
<td>S2D_A</td>
<td>cdb</td>
<td>S2D spectrum for fibre A</td>
</tr>
<tr>
<td>4</td>
<td>S2D_B</td>
<td>cdb</td>
<td>S2D spectrum for fibre B</td>
</tr>
<tr>
<td>5</td>
<td>S1D_A</td>
<td>cdb</td>
<td>S1D spectrum for fibre A</td>
</tr>
<tr>
<td>6</td>
<td>S1D_FLUXCAL_A</td>
<td>cdb</td>
<td>S1D flux calibrated spectrum for fibre A</td>
</tr>
<tr>
<td>7</td>
<td>S1D_B</td>
<td>cdb</td>
<td>S1D spectrum for fibre B</td>
</tr>
<tr>
<td>8</td>
<td>CCF_A</td>
<td>cdb</td>
<td>CCF for fibre A</td>
</tr>
</tbody>
</table>

10.12.3 Quality control

This recipe computes the following QC parameters (on the image extension n, order m, and read-out region i,j):

QC ORDERm SNR,SNR in the order m
QC ORDERm COSMIC NB,Cosmics number in the order m
QC ORDERm FLUX CORR,Flux correction for the order m
QC EXTn ROXi ROYj MAX FLUX,Max flux, raw image [ADU]
QC EXTn ROXi ROYj BIAS RON,RON[ADU] for ext 0
QC DRIFT DETn REJECTED
QC DRIFT DET1 FIRST_ORDER
QC DRIFT DET1 LAST_ORDER
QC DRIFT DET1 FLUX_RATIO
QC DRIFT DET1 FLUX_RATIO_ERR
QC DRIFT DET1 MEAN
QC DRIFT DET1 MEAN_ERR
QC DRIFT DETn SLOPE_X
QC DRIFT DETn SLOPE_X_ERR
QC DRIFT DETn SLOPE_O
QC DRIFT DETn SLOPE_O_ERR
QC DRIFT DETn CHISQ
QC SATURATION CHECK
QC BERV
QC BJD
QC BERVMAX
QC DRIFT MEAN
QC GUIDING ERROR
QC SCIRED GUIDING CHECK
QC EXPMETER LOWFLUX TIME
QC SCIRED EXPMETER CHECK
QC EXTrn BKGR MIN/MAX/MEAN
QC SCIRED DRIFT CHI2 CHECK
QC SCIRED CHECK
QC CCF RV, Radial velocity [km/s]
QC CCF RV ERROR, Error on Radial velocity [km/s]
QC SCIRED FLUX CORR CHECK, Flux correction QC
QC FLUX CORR MIN, Min of flux correction
QC FLUX CORR MAX, Max of flux correction
QC CCF FWHM, CCF FWHM [km/s]
QC CCF FWHM ERROR, CCF FWHM error [km/s]
QC CCF CONTRAST, CCF contrast %
QC CCF CONTRAST ERROR, CCF contrast error %
CCF RESIDUALS BARY, CCF residuals barycentric correction

Please note that previous list refers to the case of SCIENCE_FP. In case of SCIENCE_SKY, there is no drift computation, nor corresponding check. Moreover in that case S2D_BLAZE_THAR_FP is not a necessary input.

10.12.4 Parameters

The user may obtain brief description of the main input parameters by typing esorex -help espdr_sci_red
11 Algorithms and recipe details

In this section we describe the main algorithms implemented in the ESPRESSO pipeline recipes. Relevant data reduction parameters are typed in **bold** face. For convenience we omit the common prefix `espdr.` for the recipe description as well as the step prefix name for the algorithm description.

11.1 Algorithms

11.1.1 Master frame creation using a kappa-sigma-clipped mean/median

A kappa-sigma-clipped mean or median 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 small number of combined values a median is used rather than mean.

11.1.2 Order definition

Orders and slices are identified from ORDERDEF frames through a clumping algorithm that groups together neighbouring pixels whose values exceed a given flux threshold. The algorithm acts recursively on all pixels meeting the flux criterion within a given distance until all pixels have been classified either as belonging to a clump or as background.

Clumps are then screened against several criteria to establish whether they represent valid orders/slices: minimum length in main dispersion direction and minimum width in cross-dispersion direction.

After that, orders and slices are assigned physical numbers using input parameters specifying the number of slices per order, the physical number of the first detected order, the number of valid orders to be found, and the typical inter-slice distance on the detector.

Finally, order shapes and positions in cross-dispersion direction are fitted at several locations along the main dispersion using an appropriate analytical model (Gaussian or sum of Gaussians). Note that the Gaussian or multi-Gaussian fit is only needed to approximately measure order center. The aimed precision is of only 0.5 pixel so that the choice of the fitting function does not matter. Finally, a low-order polynomial is fitted to the Y position of each order as a function of X, where X is the main dispersion direction and Y is the cross-dispersion direction. Only order positions in Y are relevant at this stage, not their shape in the Y direction. The only purpose of these polynomials is to provide order center vs. X position to 0.5 pixel precision in order to define the window of extraction that will be used by the subsequent spectrum extraction procedure.

11.1.3 Background subtraction

The strategy to measure the diffuse background light on the detector is the following:

- Divide the detector into a grid of small regions (100x100 pixels in size)
- In each region, build the histogram of pixel values, excluding pixels falling on spectral orders
• Estimate the mode of the pixel value distribution by fitting a second-order polynomial to the histogram bins closest to the peak of the distribution. The mode of the distribution is taken as the best approximation of the local background level.

• Interpolate the background levels measured in each region over the whole detector using cubic splines.

A background map is then generated and subtracted from the raw frame.

11.1.4 Spectrum Extraction

As of today, all major fiber-fed spectrographs designed for precise radial velocity work produce relatively symmetric, round slit images on the detector (e.g. ELODIE, CORALIE, HARPS, SOPHIE). This results from the use of circular fibres and simple light injection with no image or pupil slicing at spectrograph entrance. Moreover, the curvature of echelle orders on the detector is generally small. In this case, spectrum extraction can be made by simply summing the flux in cross-dispersion direction, parallel to the CCD rows/columns. The resulting S2D spectrum is sampled on the grid of extracted pixels. Note that this extraction process has the advantage of avoiding any resampling procedure.

In the case of ESPRESSO, the situation is more complex since pupil slicing and anamorphic magnification are used to achieve high spectral resolution. The slit image on the detector is made of two ellipsoidal slices aligned in cross-dispersion direction. Depending on the exact geometry of the slit images, image quality and desired spectral resolution, it may or may not be possible to extract the spectrum by summing along the CCD rows/columns in cross-dispersion direction. Fortunately, in the optical design of ESPRESSO, the Anamorphic Pupil Slicer Unit (APSU) can be optimised to obtain a slit image tilt close to zero and ensure a precise alignment of the two slices along CCD rows/columns. The residual tilts and slice misalignments never exceed 2-3 microns anywhere on the detector, i.e. much less than the size of a pixel (10 microns). Given the minimum sampling of 2 pixels per FWHM in the ultra-high resolution mode, this means that a simple extraction procedure parallel to CCD rows/columns will decrease spectral resolution by only a few percent in the worst case, i.e. at CCD edges. We consider this as fully acceptable given the simplicity and advantages of the extraction along CCD rows/columns (e.g. no resampling).

For the modes with the two slices of an order well separated on the CCD (SINGLEHR11, SINGLEHR21 and SINGLEUHR), each slice is extracted separately and treated as separate order. This results in each physical order being represented twice in the S2D extracted spectrum.

Two spectrum extraction functions are provided in the ESPRESSO DRL: simple summation and optimal extraction following the well-known Horne method (Horne 1986, PASP 98, 609). We investigated refinements introduced by e.g. Marsh (1989) and Donati et al. (1997), but they are generally not relevant in the case of ESPRESSO because slit image tilt on the CCD is negligible and we are thus able to avoid any resampling of the data. Nevertheless, the Horne optimal extraction algorithm has been modified significantly and optimised for ESPRESSO in the following ways:

• Order profiles in cross-dispersion direction are obtained directly from spectral flat-field exposures at high SNR, which serve as empirical models. Such exposures are part of the standard calibration sets and must be obtained within 24 hours of science observations. We make use here of the high stability of ESPRESSO which guarantees that the position and shape of spectral orders remain essentially constant between calibrations and science exposures.
• If the number of hot and bad pixels does not exceed the significant fraction of the extraction window, hot and bad pixels are corrected by interpolating the order profile across them before extracting the spectra. In this way the knowledge and smoothness of the order profile can be used to optimally correct for these CCD defects.

• For exposures taken with simultaneous reference on fiber B (ThAr/comb/FP), a contamination frame is used as additional empirical model in the optimal extraction. This means that the spectrum is modelled as the sum of a scaled order profile and a scaled contamination profile. The obtained science flux is therefore corrected from direct ThAr/comb/FP contamination. As for the usual optimal extraction, the model is linear in both model parameters (science and contamination fluxes), and the solution is thus obtained by analytically solving the linear least-squares problem.

11.1.5 Flat-Fielding

The extraction procedure described above delivers a spectrum that is still affected by variable pixel-to-pixel sensitivity along the main dispersion direction. To correct for that, the spectrum is divided by the normalised extracted flat-field, obtained from co-added FLAT frames and extracted in exactly the same way as the science spectrum.

We note that this flat-fielding procedure will remove not only pixel-to-pixel sensitivity variations, but also fringing effects, at least when the science spectrum is broadly similar to the one of the flat-field lamp. This is the case for all stellar spectra at moderate to high SNR. In the case of a very low SNR spectrum dominated by sky emission lines, the correction of fringing may in principle be less accurate. However, fringing by sky lines is expected to be a very small effect at the high spectral resolution of ESPRESSO.

11.1.6 Wavelength Calibration

There are two methods of the wavelength calibration, according to the calibration source used: ThAr lamp together with Fabry-Perot or Laser Frequency Comb.

ThAr and Fabry-Perot Lamps

First the FP peaks are identified on the frame with both fibres illuminated with the Fabry-Perot lamp. Then the tables of these peaks are passed to the reduction of the frames with one of the fibres illuminated with the ThAr lamp, the other has the Fabry-Perot light for the drift calculation. Combining together the detected FP peaks and the carefully chosen ThAr lines with precise wavelengths assigned, allows to compute the precise wavelength solution per extracted order. The wavelength solution fit is done on the detected Fabry-Perot peaks, which are anchored to the precise wavelength through the known ThAr lines.

ThAr spectra are characterised by a high density of emission lines with many blends and very different intensities. Identifying and properly fitting all these lines is a challenging task, especially when aiming at a repeatability of better than 1 m/s in the wavelength calibration. Variable line intensities make it necessary to use a static ThAr table to initiate the wavelength calibration process. This table indicates the approximate position of lines. Each ESPRESSO mode has its own ThAr lines table.

ThAr lamps evolve with time, and the ageing process has several consequences on their spectra. Both line intensities and wavelengths are affected. To mitigate this, a line-by-line comparison with the static ThAr table
is performed to identify and reject individual lines potentially affected by fitting instabilities.

**Laser Frequency Comb**

Giving the previously computed wavelength solution, the Laser Frequency Comb allows to achieve a more stable and precise solution. No static table is needed to initiate the wavelength calibration process, since emission LFC lines can be searched and identified automatically thanks to the regular comb-like pattern. Then, the wavelength of each peak is determined. To do this, the DRL first computes a list of theoretical LFC wavelengths based on the known repetition rate and anchor frequency of the LFC. Then, these wavelengths are assigned to individual peaks by comparison with a previously-obtained ThAr wavelength calibration, which provides a suitable first guess for the wavelength of the LFC lines. Since the LFC does not cover the whole spectral range, the orders without LFC light have to take the solution from the ThAr method.

### 11.1.7 Instrumental Drift Measurement

The simultaneous reference fibre of ESPRESSO is used either for sky subtraction or instrumental drift correction. Instrumental drift is measured using wavelength calibration sources, either the ThAr lamp, the LFC or a stabilised Fabry-Perot etalon (FP). Drift is measured by comparing two similar S2D spectra on the simultaneous reference fibre: one that is acquired at the time of the wavelength calibration, and one that is acquired simultaneously with the science observation. The algorithm used by the DRL to compute the drift is based on the method described in Connes (1985), ApSS 110, 211, and Bouchy et al. (2001), A&A 374, 733. The drift is simply obtained from a first-order Taylor expansion of the spectrum at each individual pixel, i.e. the measured flux difference between the two spectra at pixel X is converted into a position difference using the measured spectrum derivative at pixel X. This obviously assumes that the spectrum derivative is constant over a length X that is the shift between the two spectra. Therefore, this method only works for drifts that are small relative to the FWHM of the spectral lines, i.e. typically smaller than one pixel. The pixel size in ESPRESSO is equal to 500 m/s, while the expected instrumental drifts are smaller than 1 m/s over 24 hours. The method described here is thus applicable. It has the significant advantage that no modelling of the spectrum is needed to measure the drift, since each individual pixel independently contributes to the total drift. Note that pixels are weighted according to their Doppler information content, which is proportional to the square of the spectrum derivative.

This algorithm has been successfully used with HARPS to measure drifts on ThAr spectra. For ESPRESSO a few improvements were made to the method to adapt it to LFC/FP spectra and make it more robust in general. The following aspects are addressed:

- Flux normalisation: the method only works if the two spectra are normalised to the same flux level. Thus a global flux normalisation on the S2D spectrum is performed.

- Local vs. global drift measurement: instrumental drifts may vary across the spectral format, i.e. a single, global value for the drift may be too coarse an approximation. It is possible to specify to the algorithm whether the drift computation and correction should be global, chip-by-chip or order-by-order.

With the algorithm described here, the instrumental drifts can be reliably measured to a precision of 1-2 cm/s globally (if photons allow).
11.1.8 Sky Subtraction

Sky subtraction in ESPRESSO can be made by observing in simultaneous sky mode and subtracting the sky spectrum obtained on fibre B from the science spectrum on fibre A. In practice, this process involves several steps:

- Measure the relative throughput of fibre B with respect to fibre A as a function of wavelength (in S2D format) by acquiring EFF_AB exposures, which are obtained by pointing the telescope at daylight sky. Note that the sky fills the entire fibre aperture uniformly, so that pointing/guiding errors or telescope pointing jitter have no effect on the amount of sky flux entering the spectrograph.
- Scale the S2D simultaneous sky spectrum using the previously-obtained relative efficiency curve.
- Rebin the scaled S2D sky spectrum to the same wavelength scale as fibre A. The resampling is done by spline interpolation of the cumulative flux vs. extracted pixel, ensuring flux conservation. Note that the wavelength scales of fibres A and B are extremely similar, so that the resampling will essentially consist of a small shift of the spectrum that is slowly varying along spectral orders.
- Subtract the scaled and rebinned S2D simultaneous sky spectrum from the S2D science spectrum.

The precision of sky subtraction with ESPRESSO is essentially photon- and readout-noise limited, provided the sky fibre sees the same sky as the science fibre.

11.1.9 Barycentric Correction

In the context of high-resolution astronomical spectroscopy and high-precision radial velocity measurements, it is necessary to compute to high accuracy the projection of the velocity vector of an Earth-bound observer along the line of sight to an astronomical target, at the time of the observation, as measured within the ICRS reference frame (centred on and at rest with respect to the Solar System barycenter). The computation should take into account both the Earth’s orbital motion around the Sun and the observer’s motion due to Earth rotation. For ESPRESSO a new code for this computation will be developed, based on recent Solar System ephemerides produced at Institut de Mécanique Céleste et de Calcul des Ephémérides (IMCCE) in Paris.

The algorithm:

- Computes the observer’s projected velocity in the direction of the target in the ICRS reference frame at the time of the observation with the accuracy of at least 1 cm/s.
- Computes the barycentric time of light arrival, i.e. the time at which the light from the target reaches the Solar System barycenter given its detection by the Earth-bound observer at the time of the observation with the accuracy of at least 1s.
- Computes an upper bound to the maximum value of the barycentric correction over one year for any given target.

Input parameters:
• Right ascension of target in decimal hours, ICRS system, epoch 2000.0

• Declination of target in decimal degrees, ICRS system, epoch 2000.0

• Proper motion of target in right ascension (\(\mu_\alpha \cos \delta\)) in arcsec/year, ICRS system

• Proper motion of target in declination in arcsec/year, ICRS system

• Date of observation in UTC (year, month, day, decimal hour)

• Longitude of observer in decimal degrees

• Latitude of observer in decimal degrees

• Altitude of observer above sea level in km

Output parameters:

• Velocity component of observer projected onto target line of sight in km/s, ICRS system (abridged as BERV for Barycentric Earth Radial Velocity)

• Total velocity of observer in km/s, ICRS system

• Barycentric time of light arrival as UTC Julian date

• Yearly maximum value of BERV (upper bound)

Step-by-step description:

• Compute Julian date from given date and time of observation

• Compute ICRS target coordinates at the time of observation (i.e. take proper motion into account)

• Compute observer’s velocity vector with respect to geocenter at the time of observation, ICRS system (correct for precession and nutation)

• Compute Earth’s orbital velocity vector at the time of observation, ICRS system

• Combine rotational and orbital motion into total velocity vector

• Project velocity vector onto target line of sight to obtain BERV

• Compute barycentric time of light arrival

• Compute yearly upper bound to BERV
11.1.10 Order Rebinning and Merging

The rebinning and merging process to create a S1D spectrum from the S2D spectrum is as follows:

- Define a uniform wavelength grid with a constant wavelength step that is close to the average pixel size in wavelength units.
- For each spectral order, build the cumulative flux distribution vs. extracted pixel.
- Interpolate the cumulative flux function onto the uniform wavelength grid using cubic splines.
- Build the resampled spectrum by differentiating the interpolated cumulative flux.
- Merge resampled spectral orders, computing the weighted average of rebinned pixels where spectral orders overlap.

This technique is chosen because it conserves the integrated flux within any two wavelengths of the original pixel grid. Moreover, cubic splines are well suited to ESPRESSO spectra because of the well-sampled PSF of the instrument (4 pixels FWHM in singleHR mode), which ensures that the spectrum derivatives can be numerically estimated in a reliable way.

11.1.11 Radial Velocity Computation

The ESPRESSO DRL implements a cross-correlation module that computes the cross-correlation function (CCF) of a S2D spectrum with respect to a binary template (mask) of a given spectral type. The radial velocity (RV) is then obtained from a Gaussian fit to the CCF. This is the technique that has been successfully used on the ELODIE, CORALIE, HARPS, SOPHIE and HARPS-N spectrographs (see Baranne et al. 1996, A&AS 119, 373, and Pepe et al. 2002, A&A 388, 632). One of its main advantages is that CCFs can be computed in an automatic way with only a few line masks at hand. Line masks are simply lists of central wavelengths and depths of spectral lines, and can be created for various spectral types. We note here that the CCFs of slowly-rotating stars are extremely well approximated by Gaussian profiles with a flat continuum. However, the particular fitting function does not matter much; the crucial aspect is to fit the CCFs of a given star with always the same function to avoid systematic effects on the derived radial velocities.

The main steps of the algorithm are:

- Compare the global flux distribution in the S2D spectrum to a static flux template that approximately corresponds to the spectral type of the star. The S2D flux is scaled accordingly to match the flux distribution of the template. In this way, spectra of any given star are always brought to the same flux distribution, which ensures that variable atmospheric conditions will not induce systematic effects in the CCF computation.
- Shift the wavelength scale of the S2D spectrum to the Solar System barycenter using the barycentric correction.
- Define a uniform radial velocity grid that is approximately centered on the radial velocity of the star.
• For a given RV value in the grid, shift the line mask by the corresponding Doppler shift, project the line mask onto the S2D spectrum using a specified line width (about one pixel), and sum the S2D flux that goes through the so-defined mask holes. The flux from partial pixels is computed via simple linear interpolation. The sum is actually a weighted sum, using line depths as weights to optimally extract the Doppler information. During this process, the S2D spectrum is locally blaze-corrected to remove any continuum slope around spectral lines. This produces one point of the CCF.

• Loop over all RV values.

• Fit a Gaussian profile to the CCF to derive RV, FWHM and contrast.

Note that, by construction, CCFs are simply co-added spectral lines in velocity space, weighted by their depth and continuum flux. As such they can be considered as a master spectral line for the star.

11.1.12 Error and Bad Pixel Propagation

S2D Spectra
All significant error sources are propagated through the pipeline using the standard error propagation formulae. We list below the noise sources that are taken into account in the science reduction cascade producing the S2D spectra:

• Detector readout noise
• Detector dark current noise
• Total photon noise on science fiber (science target + diffuse background)
• Flat-fielding noise
• Sky noise if applicable (simultaneous sky mode)

Error propagation for S2D spectra is relatively straightforward since there is no resampling step in the process. Extracted pixels remain independent of each other.

If not too many, bad pixels and hot pixels will be ignored during the spectrum extraction process. The empirical, high-SNR extraction profile will be interpolated across those pixels, so that the extracted science spectrum is effectively corrected. The presence of bad/hot pixels will be reported in the quality maps associated to science products; they are therefore traceable throughout the reduction chain.

S1D Spectra
The resampling process that is necessary to generate S1D spectra from S2D spectra inevitably introduces correlations between adjacent rebinned pixels in the S1D spectra. The proposed strategy here is to propagate errors in the usual way to the S1D spectrum, but to also provide an additional array containing the correlation factor of each rebinned pixel. This factor is the quadratic sum of the relative contributions of the original pixels to the rebinned pixel, considering that a maximum of two original pixels contribute to the rebinned pixel. In this way we believe all the information on noise properties is conveyed into the S1D spectrum. We note however that
the existence of correlations makes the use of the S1D spectrum non-trivial for science purposes if a rigorous
treatment of the noise is required. It is up to the user to understand how to deal with this issue, e.g. when
performing least-squares fitting of models to the S1D spectrum.

In general, the use of the S2D spectrum is recommended whenever possible to avoid correlations between
adjacent data points, which are unavoidable as soon as some resampling is performed.

Cross-Correlation Functions

Uncertainties on the CCF data points are obtained by propagation of S2D error maps through the cross-correlation
stage, which is a simple additive process (fluxes from many spectral lines are co-added). Finally, an estimate
of the radial velocity uncertainty is obtained by converting CCF flux errors into RV errors using the measured
CCF derivative, as described in Boisse et al. (2010), A&A 523, 88 (appendix).

11.2 Recipes

An overview of the data reduction cascade is given in Section9.

11.2.1 mbias

The recipe generates a master bias frame from a set of minimum 5 raw bias frames. All the input raw frames
are stacked and a sigma-clipping algorithm is applied in order to remove the cosmics. Then, the overscan is
removed from the master bias image and the rest is save as a master bias residuals. These residuals are applied
in several recipes in the reduction chain. The BIAS RON and mean and the overscan RON are computed on the
master bias frame and saved in the FITS header as QC KWs. The mean and stdev are computer on the master
bias residuals frame and saved in the FITS header as QC KWs.

11.2.2 mdark

The recipe creates a master dark frame from a set of minimum 5 dark frames corrected for the bias. All the
input raw frames are stacked and a sigma-clipping algorithm is applied to remove the cosmics. The cosmics
rates by extension (bleu and red) are saved in the FITS header as QC KWs. The hot pixels are detected via the
sigma-clipping algorithm and saved in the hot_pixels frame. Their number per detector output is saved in the
FITS header as QC KW. The mean dark current is computed for each detector output and saved in the FITS
header as QC KW.

11.2.3 led_ff

The recipe detects the bad pixels and computes the conversion factor for each detector output. For this at least
two sets with different exposure times of minimum 5 frames each are needed. The frames are the images of the
detector fully illuminated by a LED. The recipe first removes the bias and the dark current and then stacks the
frames with the same exposure time. The cosmics are removed via the sigma-clipping algorithm. The bad pixels
are detected via checking of the linearity of their behaviour with respect to the exposure time. Their number
is saved in the FITS header as QC KW. The conversion factor is computed within a window defined for each
detector output and saved in the FITS header as QC KW. The raw frames are checked against the saturation and the corresponding QC KW is set. The same is done for the minimal flux. The computed conversion factor is checked against the theoretical value taken from the FITS header of the raw frame.

11.2.4 orderdef

The recipe detects the orders position in the image. The input frames are two flat images: one for fibre A and one for fibre B. Each input raw frame is checked for the saturation with QC CHECK set accordingly and cleaned from the CCD signature (bias, dark current, gain). Than the orders detection is performed:

- For each pixel, decide if it belongs to an order or to the background
- Run the clumps detection algorithm, which marks the pixels belonging to the same clump
- Detect which of the clumps are valid orders
- Number the valid orders
- Fit the valid orders position
- Mark the orders position on the ORDER_TABLE products
- Save the coefficients of the fit in the 2nd extension of the ORDER_TABLE products

11.2.5 mflat

The recipe generates the extracted flat-field, blaze function and the order profile. Recipe steps:

- Check the raw frame for saturation
- Remove the detector signatures
- Measure and subtract inter-order background
- Perform optimal extraction of orders for both fibres
- Compute the blaze function
- Correct the blaze function for spectral energy distribution of the source and instrumental efficiency by fitting the flux distribution at blaze peak.
- Save the QC KWs in the FITS header of the products: max flux per detector output, background min, max and mean per extension (blue and red), number of removed cosmics per order, SNR per order, RMS of the flat per order, QC CHECKS
- Save the ORDER_PROFILE, the FLAT and the BLAZE for each fibre
11.2.6 cal_contam

The recipe generates a contamination frame and checks the contamination level on science fibre. Recipe steps:

- Check the raw frame for saturation
- Remove the detector signatures
- Measure and subtract inter-order background
- Perform optimal extraction of orders for both fibres
- Correct the flat-field
- Measure the maximum contamination level in extracted spectrum and compare it with the specified threshold
- Save the contamination frame with QC KWs

11.2.7 cal_eff_ab

The recipe computes the relative efficiency between sky and science fibres vs. wavelength. Recipe steps:

- Check the raw frame for saturation
- Remove the detector signatures
- Measure and subtract inter-order background
- Perform optimal extraction of orders for both fibres
- Correct the flat-field
- Divide science spectrum by reference spectrum and fit a low-order polynomial across each order
- Save the relative efficiency frame with QC KWs

11.2.8 wave_FP

Recipe detects the Fabry-Perot peaks. Recipe steps:

- Check the raw frame for saturation
- Remove the detector signatures
- Perform optimal extraction of orders for both fibres
- Correct the flat-field and the blaze
• Search for the FP lines
• Fit the FP lines
• Check if there is no lines missing
• Save the FP lines table and S2D, S2D_BLAZE products with QC KWS

11.2.9 wave_THAR

Recipe performs the wavelength calibration using the ThAr lamp and Fabry-Perot spectra. Recipe steps:

• Remove the detector signatures
• Perform optimal extraction of orders for both fibres
• Correct the flat-field and the blaze
• Fit the ThAr lines provided in a static table
• Measure drift and apply it to the FP lines table (product of the wave_FP recipe)
• Check if there is no FP lines missing
• Compute the wavelength for all the FP lines, using ThAr lines
• Fit D lambda
• Fit the wavelength solution
• Fit the ThAr raw lines
• Save the products with QC KWS

11.2.10 wave_LFC

Recipe performs the wavelength calibration using the Laser Frequency Comb spectrum. Recipe steps:

• Remove the detector signatures
• Perform optimal extraction of orders for both fibres
• Correct the flat-field and the blaze
• Fit the ThAr lines provided in a static table
• Search for the LFC lines
• Fit the LFC lines
• Assign wavelengths to the LFC lines
• Fit the wavelength solution
• Complete the wavelength solution with the ThAr one for orders with no LFC flux
• Save the products with QC KWs

### 11.2.11 cal_flux

The recipe measures the absolute efficiency curve. Recipe steps:

• Check the raw frame for saturation
• Remove the detector signatures
• Measure and subtract inter-order background
• Perform optimal extraction of orders for both fibres
• Correct the flat-field
• Assign wavelength solution to the spectrum
• Rebin and merge the orders
• Convert S1D spectrum from detected photoelectrons to physical units (erg/s/cm²/A)
• Measure and retrieve the flux of the spectrophotometric standard star, and compute absolute efficiency at the reference wavelengths
• Interpolate efficiency measurements onto the S1D wavelength scale using cubic splines
• Save the absolute efficiency frame with QC KWs

### 11.2.12 sci_red

The recipe performs the science reduction. Recipe steps:

• Check the raw frame for saturation
• Remove the detector signatures
• Measure and subtract inter-order background
• Perform optimal extraction of orders for both fibres
• Correct the flat-field
• Assign wavelength solution to the spectrum
• In simultaneous sky mode:
- Scale and rebin sky spectrum
- Subtract sky

• In simultaneous reference mode:
  - Measure instrumental drift
  - Correct instrumental drift

• Compute barycentric correction using FITS header information
• Correct wavelength solution from BERV, shifting it to the barycenter of the SS
• Rebin and merge orders
• Convert S1D flux into physical units
• Flux calibrate S1D spectrum (with and without sky subtraction) using absolute efficiency curve
• Measure fiber centering on integrated guiding image and perform quality control based on specified tolerances
• Measure flux-weighted mid-exposure time on exposure-meter data and perform quality control
• Filter residual cosmic hits from S2D spectrum
• Correct flux distribution in S2D spectrum
• Compute the radial velocity
• Compute the CCF bisector
• Save the S2D, S1D and CCF products with QC KWs
A  Installation

This chapter gives generic instructions on how to obtain, build and install the ESPRESSO pipeline. Even if this chapter is kept as up-to-date as much as possible, it may not be fully applicable to a particular release. This might especially happen for patch releases. One is therefore advised to read the installation instructions delivered with the ESPRESSO pipeline distribution kit. These release-specific instructions can be found in the file README located in the top-level directory of the unpacked ESPRESSO pipeline source tree. The supported platforms are listed in Section A.1. It is recommended reading through Section A.2.2 before starting the installation.

A bundled version of the ESPRESSO pipeline with all the required tools and an installer script is available from www.eso.org/pipelines, for users who are not familiar with the installation of software packages.

A.1  Supported platforms

The utilisation of the GNU build tools should allow to build and install the ESPRESSO pipeline on a variety of UNIX platforms, but it has only been verified on the VLT target platforms:

- Linux (glibc 2.1 or later),
- Mac,

using the GNU C compiler (version 7.3 or newer). Gasgano is installed by the install_pipeline script and is supported provided the user has a proper installation of the Java Development Kit (version 1.8.0 or newer) and has set the JAVA_HOME environment variable properly during the kit installation. It is not sufficient to have the Java Realtime Environment to have successfully installed the libgasganocpl.* libraries needed to properly interface Gasgano with the installed espdrni recipes. We recommend the user to look carefully at the log produced during installation and possible warnings.

A.2  Building the ESPRESSO pipeline

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

A.2.1  Requirements

To compile and install the ESPRESSO pipeline one needs:

- the GNU C compiler (version 7.3 or later),
- the GNU gzip data compression program,
- a version of the tar file-archiving program, and,
- the GNU make utility.
An installation of the Common Pipeline library (CPL) must also be available on the system. Currently the CPL version 7.1 or newer is required. The CPL distribution can be obtained from www.eso.org/cpl.

In order to run the ESPRESSO pipeline recipes a front-end application is also required. Currently there are two such applications available, a command-line tool called EsoRex and the Java based data file organizer, Gasgano, which provides an intuitive graphical user interface (see Section 5.2, page 20). At least one of them must be installed. The EsoRex and Gasgano packages are available at www.eso.org/cpl and www.eso.org/gasgano respectively.

For installation instructions of any of the additional packages mentioned before please refer to the documentation of these packages.

### A.2.2 Compiling and installing the ESPRESSO pipeline

The ESPRESSO pipeline distribution kit 1.1.0 contains:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>espdr-pipeline–manual-1.0.pdf</td>
<td>The ESPRESSO pipeline manual</td>
</tr>
<tr>
<td>install_pipeline</td>
<td>Install script</td>
</tr>
<tr>
<td>cfitsio3360.tar.gz</td>
<td>CFITSIO 3360</td>
</tr>
<tr>
<td>cpl-7.1.tar.gz</td>
<td>CPL 7.1</td>
</tr>
<tr>
<td>esorex-3.13.tar.gz</td>
<td>esorex 3.13</td>
</tr>
<tr>
<td>gasgano-2.4.8.tar.gz</td>
<td>GASGANO 2.4.8</td>
</tr>
<tr>
<td>espmni-1.1.0.tar.gz</td>
<td>ESPRESSO 1.1.0</td>
</tr>
<tr>
<td>espmni-calib-1.1.0.tar.gz</td>
<td>ESPRESSO calibration files 1.1.0</td>
</tr>
</tbody>
</table>

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the ESPRESSO pipeline recipes 1.1.0 package. It can be any directory of your choice but not:

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

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

3. Verify the checksum value of the tar file with the cksum command.

4. Unpack using the following command:

   ```bash
   tar -xvf espdr-kit-1.1.0.tar
   ```

   Note that the size of the installed software (including Gasgano) together with the static calibration data is about 27Mb.
5. Install: after moving to the top installation directory, `cd espdr-kit-1.1.0`

   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/`

   (beware: the execution may take a few minutes on Linux and several minutes on SunOS).

   By default the script will install the ESPRESSO 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 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).

   Alternatively, it is possible to perform a manual installation (experienced users only): the README file located in the top installation directory contains more detailed information about a step-by-step installation.