## Authors

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Modigliani, M. Neeser</td>
<td>ESO</td>
</tr>
<tr>
<td>E. Wiezorrek, Y. Cao</td>
<td>MPE</td>
</tr>
</tbody>
</table>
## Change Record from previous Version

<table>
<thead>
<tr>
<th>Release</th>
<th>Affected Section(s)</th>
<th>Changes/Reason/Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.0</td>
<td>All</td>
<td>First public release</td>
</tr>
<tr>
<td>1.1.3</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>1.2.0</td>
<td>Sec 8, 9, 12</td>
<td>Aligned to pipeline update</td>
</tr>
<tr>
<td>1.2.5</td>
<td>Sec 1, 5, 8, 12</td>
<td>Improve document</td>
</tr>
<tr>
<td>1.4.1</td>
<td>Sec 1, 5, 7, 8, 9, 11,12, 13, App A</td>
<td>Improve document</td>
</tr>
<tr>
<td>1.4.4</td>
<td>Sec 5, 8, 12</td>
<td>Align to pipeline update</td>
</tr>
<tr>
<td>1.5.0</td>
<td>Sec 1, 4, 5, 8, 12</td>
<td>Align to pipeline update</td>
</tr>
<tr>
<td>1.5.3</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>1.6.0</td>
<td>Sec 5</td>
<td>Align to pipeline update</td>
</tr>
</tbody>
</table>
# Contents

1 Introduction
   1.1 Purpose ................................................................. 7
   1.2 Scope ................................................................. 7
   1.3 Acknowledgements .................................................. 7
   1.4 Stylistic conventions ............................................... 8
   1.5 Notational Conventions ............................................ 8

2 Related Documents ...................................................... 9

3 Definitions, Acronyms and Abbreviations .......................... 10

4 Overview ........................................................................ 11

5 What's new in pipeline release 1.6.0 ................................. 12

6 The ERIS Instrument ...................................................... 13
   6.1 ERIS-SPIFFER, Spectrograph ..................................... 13
   6.2 ERIS-NIX, Infrared Imager ........................................ 14
   6.3 ERIS and the Adaptive Optics Facility .......................... 14

7 Quick Start ................................................................. 15
   7.1 The ERIS Pipeline Recipes ......................................... 15
   7.2 Running the ERIS Pipeline Recipes ............................... 15
      7.2.1 Getting Started with EsoRex ................................. 15
   7.3 Data Organization ................................................... 16
   7.4 Example of data reduction using EsoRex ....................... 21

8 Known Issues ............................................................. 27
   8.1 Use ESO Reflex ....................................................... 27
   8.2 Combining Multiple Cubes (possibly observed with different OBs) ...................... 27
   8.3 Correcting possible residual minor shifts of the observed object .......................... 28
   8.4 Combining Multiple Cubes ........................................ 28
8.5 Refining the Wavelength Calibration Using Sky Lines ............................................. 28
8.6 Missing BPM_DARK in Reflex DataOrganiser ......................................................... 29
8.7 Wiggles Visible in Under-Sampled Data Cubes ....................................................... 29
8.8 Flux calibrated spectrum defined on a wavelength range shorter than the one of the extracted spectrum ................................................................. 32
8.9 Possible failure of recipe eris_ifu_jitter in case --crea-phase3=true and --sky_tweak=1 ........ 32
8.10 Obtaining proper flux calibrated results ............................................................... 33
8.11 Determination of response and spectrum flux calibration based on photometric standard stars not part of the calibration plan ......................................................... 33
8.12 Possible failure of recipe eris_ifu_jitter in case --mask_method is set to ‘fit’ .................... 34

9 ERIS Data Description ........................................................................................................ 35
9.1 Darks .......................................................................................................................... 36
9.2 Linearity frames .......................................................................................................... 36
9.3 North-South test data ............................................................................................... 37
9.4 Flats ............................................................................................................................ 37
9.5 Arc lamp frames ........................................................................................................ 38
9.6 Flux standard star frames .......................................................................................... 38
9.7 PSF standard star frames ........................................................................................... 38
9.8 Other standard star frames ....................................................................................... 39
9.9 Pupil monitoring frames ........................................................................................... 39
9.10 Science observations ............................................................................................... 39

10 Static Calibration Data ..................................................................................................... 41
10.1 First Wave Fit ........................................................................................................... 41
10.2 Reference Line Catalogue ......................................................................................... 41
10.3 Wavelength Setup Table .......................................................................................... 42
10.4 Catalog of Flux Standards Stars ............................................................................... 42
10.5 Atmospheric extinction table .................................................................................... 42
10.6 Efficiency Windows .................................................................................................. 42
10.7 Fit Areas .................................................................................................................... 43
10.8 Fit Areas .................................................................................................................... 43
10.9 High Abs Regions .......................................................... 43
10.10 Quality Areas ............................................................. 43
10.11 Response Windows ..................................................... 43
10.12 Catalog of fit points to fit the Response ......................... 44
10.13 Catalogue of Telluric models ....................................... 44

11 Data Reduction ............................................................. 45
11.1 The ERIS Data Reduction Pipeline ................................. 45
11.2 Data reduction overview .............................................. 45
11.3 Required input data .................................................... 47
11.4 Reduction cascade ....................................................... 48

12 Recipe Reference .......................................................... 50
12.1 Common Recipe Parameters ........................................ 50
12.2 eris_ifu_dark ............................................................. 51
  12.2.1 Description ......................................................... 51
  12.2.2 Input Frames ....................................................... 51
  12.2.3 Product Frames .................................................... 51
  12.2.4 Recipe Parameters ................................................ 52
  12.2.5 Quality Control Parameters .................................... 54
12.3 eris_ifu_detlin ............................................................ 55
  12.3.1 Description ......................................................... 55
  12.3.2 Input Frames ....................................................... 55
  12.3.3 Product Frames .................................................... 55
  12.3.4 Quality Control Parameters .................................... 55
  12.3.5 Recipe Parameters ................................................ 55
12.4 eris_ifu_distortion ...................................................... 56
  12.4.1 Description ......................................................... 56
  12.4.2 Input Frames ....................................................... 56
  12.4.3 Product Frames .................................................... 57
  12.4.4 Quality Control Parameters .................................... 57
12.4.5 Recipe Parameters .......................................................... 57
12.5 eris_ifu_flat ................................................................. 59
  12.5.1 Description ............................................................. 59
  12.5.2 Input Frames .......................................................... 59
  12.5.3 Product Frames ....................................................... 59
  12.5.4 Quality Control Parameters ....................................... 59
  12.5.5 Recipe Parameters .................................................. 59
12.6 eris_ifu_wavecal .......................................................... 60
  12.6.1 Description ............................................................. 60
  12.6.2 Input Frames .......................................................... 60
  12.6.3 Product Frames ....................................................... 61
  12.6.4 Quality Control Parameters ....................................... 61
  12.6.5 Recipe Parameters .................................................. 62
12.7 eris_ifu_stdstar ............................................................ 63
  12.7.1 Description ............................................................. 63
  12.7.2 Input Frames .......................................................... 63
  12.7.3 Product Frames ....................................................... 64
  12.7.4 Quality Control Parameters ....................................... 64
  12.7.5 Recipe Parameters .................................................. 66
12.8 eris_ifu_jitter .............................................................. 67
  12.8.1 Description ............................................................. 67
  12.8.2 Input Frames .......................................................... 68
  12.8.3 Product Frames ....................................................... 68
  12.8.4 Recipe Parameters .................................................. 70
  12.8.5 Quality Control Parameters ....................................... 75
12.9 eris_ifu_combine hdrl ................................................... 75
  12.9.1 Description ............................................................. 75
  12.9.2 Input Frames .......................................................... 76
  12.9.3 Product Frames ....................................................... 76
  12.9.4 Recipe Parameters .................................................. 76
12.9.5 Quality Control Parameters .............................................. 77
12.10 eris_ifu_combine ................................................................. 78
  12.10.1 Description ................................................................. 78
  12.10.2 Input Frames ............................................................... 78
  12.10.3 Product Frames ............................................................ 78
  12.10.4 Recipe Parameters ........................................................ 78
  12.10.5 Quality Control Parameters ........................................... 79

13 Algorithms ............................................................................ 80
  13.1 General Algorithms .......................................................... 80
  13.2 Recipes Algorithms .......................................................... 80
    13.2.1 Dark data reduction ....................................................... 80
    13.2.2 Linearity data reduction ............................................... 80
    13.2.3 North-South (distortion) data reduction .......................... 80
    13.2.4 Flat data reduction ....................................................... 81
    13.2.5 Wavelength calibration .................................................. 81
    13.2.6 OH based Wavelength calibration ................................. 82
    13.2.7 Creation of a 3D data cube ............................................ 82
    13.2.8 Correction of the Differential Atmospheric Refraction ....... 83
    13.2.9 Sky model ................................................................. 84
    13.2.10 Cube resampling ....................................................... 85
    13.2.11 Instrument Response computation ................................ 87
    13.2.12 Flux calibration ......................................................... 89
    13.2.13 Strehl computation ..................................................... 89
    13.2.14 Standard star position detection ................................. 90

A Installation .............................................................................. 91
  A.1 System Requirements ......................................................... 91
  A.2 Installing the ERIS Pipeline .................................................. 91
    A.2.1 Build Requirements ...................................................... 91
1 Introduction

1.1 Purpose

The ERIS-SPIFFIER pipeline is a subsystem of the VLT Data Flow System (DFS). It is used in two operational environments, for the ESO Data Flow Operations (DFO), and the Paranal Science Operations (PSO), in the quick-look assessment of data, in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. Additionally, the ERIS-SPIFFIER pipeline recipes are made public to the user community, to allow a more personalised processing of the data of the instrument. The purpose of this document is to describe a typical ERIS-SPIFFIER data reduction sequence with the ERIS-SPIFFIER pipeline. This manual is a complete description of the data reduction recipes implemented by the ERIS-SPIFFIER pipeline, reflecting the status of the ERIS-SPIFFIER pipeline as of version 1.6.0.

1.2 Scope

This document describes the ERIS-SPIFFIER pipeline used at ESO to assess instrument and data quality control.

Updated versions of the present document may be found on [?]. For general information about the current instrument pipelines status we remind the user of [?]. Additional information on CFITSIO, the Common Pipeline Library (CPL) and EsoRex can be found respectively at [?], [?], [?]. The Reflex front-ends is described in [?] and a description of the instrument can be found in [?]. The ERIS instrument user manual is in [?], the calibration plan in [?].

1.3 Acknowledgements

The ERIS-SPIFFIER pipeline was designed, implemented and verified by Erich Wiezorrek, Alex Agudo and Yixian Cao of the Max Planck Institute for Extraterrestrial Physics of the ERIS consortium. Kateryna Kravchenko participated in the ERIS commissioning and contributed to the pipeline verification.

Mark Neeser, from the ESO Science Data Quality group, led this pipeline project since its beginning and contributed to its design and overall scientific verification.

Isabelle Percheron, from the ESO Science Data Quality group, is responsible of the integration of the pipeline in the Quality Control environment, and from early commissioning, provided valuable feedback on the data and its quality control.

Andrea Modigliani, from the Science Operations Software Pipeline Systems Group, followed since its beginning the development of the pipeline to ensure that the algorithms are properly implemented and that the code is robust and easy to maintain. He implemented the ESOREflex workflow, added quality control parameters, contributed to the implementation and verification of several functionalities used to reduce observations, updated science products to be compliant with the phase3 standard, provided support for the commissioning runs, pipeline operations and contributed to the documentation.

Lars Lundin, from the Science Operations Software Pipeline Systems Group, contributed to improve software maintainability and portability and implementing several recipe regressions tests.
This release and documentation includes improvements to address feedback from scientists and users. We thank Julia Bodensteiner, an ESO fellow during the period of ERIS commissioning and early operations, who provided several useful suggestions to improve the SPIFFIER data reduction.

1.4 Stylistic conventions

Throughout this document the following stylistic conventions are used:

- **bold** in text sections for commands and other user input which has to be typed as shown
- *italics* in the text and example sections for parts of the user input which have to be replaced with real contents
- `teletype` in the text for FITS keywords, program names, file paths, and terminal output, and as the general style for examples, commands, code, etc

In example sections expected user input is indicated by a leading shell prompt.

In the text **bold** and *italics* may also be used to highlight words.

1.5 Notational Conventions

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


## 3 Definitions, Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalibDB</td>
<td>Calibration Database</td>
</tr>
<tr>
<td>CPL</td>
<td>Common Pipeline Library</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge Coupled Device</td>
</tr>
<tr>
<td>DFS</td>
<td>Data Flow System</td>
</tr>
<tr>
<td>DRS</td>
<td>Data Reduction System</td>
</tr>
<tr>
<td>ESO</td>
<td>European Southern Observatory</td>
</tr>
<tr>
<td>EsoReX</td>
<td>ESO Recipe Execution Tool</td>
</tr>
<tr>
<td>FITS</td>
<td>Flexible Image Transport System</td>
</tr>
<tr>
<td>FOV</td>
<td>Field Of View</td>
</tr>
<tr>
<td>GSL</td>
<td>GNU Scientific Library[?]</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HDRL</td>
<td>High Level Data Reduction Library</td>
</tr>
<tr>
<td>LSF</td>
<td>Line Spread Function</td>
</tr>
<tr>
<td>ERIS</td>
<td>Enhanced Resolution Imager and Spectrograph</td>
</tr>
<tr>
<td>OB</td>
<td>Observation Block</td>
</tr>
<tr>
<td>pixel</td>
<td>picture element (of a raster image)</td>
</tr>
<tr>
<td>PSF</td>
<td>Point Spread Function</td>
</tr>
<tr>
<td>QC</td>
<td>Quality Control</td>
</tr>
<tr>
<td>SDP</td>
<td>Science Data Product</td>
</tr>
<tr>
<td>SOF</td>
<td>Set Of Frames</td>
</tr>
<tr>
<td>TBD</td>
<td>To be defined</td>
</tr>
<tr>
<td>TBC</td>
<td>To be confirmed</td>
</tr>
<tr>
<td>VLT</td>
<td>Very Large Telescope</td>
</tr>
<tr>
<td>WCS</td>
<td>World Coordinate System</td>
</tr>
</tbody>
</table>
4 Overview

In collaboration with instrument consortia, the Data Flow Systems Department (DFS) of the Data Management and Operation Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

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

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

**Science product creation**: using pipeline-generated master calibration products, science products are produced for the supported instrument modes. The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

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

ESO offers three front-end applications for launching pipeline recipes; namely EsoReflex [?], Gasgano [?] and EsoRex [?]. These applications can also be downloaded separately from www.eso.org/reflex, www.eso.org/gasgano, www.eso.org/cpl/esorex.html respectively.

The ERIS instrument and the different types of ERIS raw frames and auxiliary data are described in Sections 6, 9.

A brief introduction to the usage of the available reduction recipes using EsoRex is presented in Section 7.1. In section 8 we summarise known data reduction problems and solutions, if available.

An overview of the data reduction, what are the input data, and the recipes involved in the calibration cascade is provided in section 11.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in section 12.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 13.

In Appendix A the installation of the eris pipeline recipes is described and in section 3 a list of used abbreviations and acronyms is given.
5 What’s new in pipeline release 1.6.0

This pipeline release includes the following improvements, several useful to desktop users:

- The science recipe data products are compliant with the ESO phase3 standard. As part of this added the `crea-phase3` parameter to enable the generation of phase3 standard data products and the parameter `chop-nan` to remove the planes that are dominated by NAN pixels from the product data cubes. As part of this activity a number of bugs have been fixed in particular on the FITS format of the jitter recipe data products.

- Added the recipe `eris_ifu_combine`, combining 3D data cubes with an algorithm similar to the one implemented in the SINFONI pipeline. With respect to the recipe `eris_ifu_combine_hdrl`, `eris_ifu_combine` requires less time to perform the same task (because it does not need to do an initial frame resampling required by the HDRL based implementation), and offers additional functionality for example to clip outliers. For details the user should read section 12.10.

- Set proper values of RA, DEC in each single 3D data cube product.

- Fixed ESOReflex problems reducing flux standard star data and broadcasting their response product to actors following in the reduction chain.

- Added EG21 entry to the `FLUX_STD_CATALOG` table catalog list.

- Improved Strehl computation robustness and fixed occasional problems determining the efficiency.

- Reduced the RAM used by the recipe `eris_ifu_detlin`, to enable reduction within 12 GB of RAM.

- Added a dedicated workflow parameter to control the interactivity of calibrations actors.

- Updated documentation.

- Implemented the recipe `eris_ifu_pupil` to enable operations to monitor the position of the telescope pupil with ERIS-SPIFFIER data.

- Added more quality control parameters and fixed a number of problems reported by operations.
6 The ERIS Instrument

ERIS is part of a new generation of technology for ESO’s Very Large Telescope (VLT) located at the Paranal Observatory in Chile. The Enhanced Resolution Imager and Spectrograph (ERIS) instrument combines a general-use infrared imager and integral field spectrograph with the world-class adaptive optics installed on the VLT’s Unit Telescope 4 (Yepun).

The versatility of ERIS lends itself to many fields of astronomical research and it aims to take the sharpest images obtained to date using a single 8.2-metre class telescope once operational. ERIS is expected to greatly contribute to probing of distant galaxies, the Galactic Centre, our Solar System and exoplanets.

The new, more sensitive technology of ERIS will succeed the successful NACO and SINFONI instruments, which have been used on the VLT since the early 2000s. ERIS is planned to work for 10 years following first light.

6.1 ERIS-SPIFFER, Spectrograph

The spectrograph of ERIS, named ERIS-SPIFFIER, is a refurbished version of SINFONI’s SPIFFI (SPectrometer for Infrared Faint Field Imaging) instrument. The ERIS-SPIFFIER, as an integral field spectrograph, works in much the same way as SPIFFI. Each pixel of the spectrograph is capable of collecting a full spectrum, which
is the intensity of each wavelength measured by the instrument. The data collected is multidimensional and contains a lot of information for astronomers to analyse. This type of spectrograph allows astronomers to determine the relative position in space of each part of the target object. Combined with the infrared observing wavelength, this type of spectroscopy can for example allow astronomers to study the spin, structure, and content of high-redshift galaxies. A higher resolution grating on the spectrograph, used to split incoming light to capture a spectrum, works similarly to how a prism splits light into different colours.

6.2 ERIS-NIX, Infrared Imager

The imager on ERIS is the Near Infrared Camera System (NIX), the next generation of infrared imager for the VLT. Astronomers are able to use two main modes for the imager. One mode is direct imaging of astronomical objects. Another mode is coronagraphy, an observing technique that works by suppressing the direct light of a star. This technique on ERIS-NIX allows astronomers to observe exoplanets and discs of gas and dust around young stars.

ERIS-NIX is capable of collecting data in near-infrared between 3000 and 5000 nanometers, which are particularly difficult wavelengths to observe from the ground due to excess light interference from the sky. As such, ERIS is one of the few ground-based instruments with this wavelength coverage on an 8-metre class telescope and observing at these wavelengths allows for a new and unique view of exoplanets and galaxies.

6.3 ERIS and the Adaptive Optics Facility

ERIS is also equipped with an adaptive optics module that works together with the Adaptive Optics Facility (AOF) installed in 2017 on Yepun. Adaptive optics make real-time corrections for atmospheric turbulence that distorts starlight to allow for much sharper image quality, and are an important part of modern ground-based astronomical observations. The AOF capabilities in conjunction with the module allows ERIS to capture the sharpest data possible with the instrument and cover more of the sky than earlier VLT instruments.
7 Quick Start

7.1 The ERIS Pipeline Recipes

The ERIS-SPIFFIER pipeline reduction of imaging data chain uses the following recipes:

- `eris_ifu_dark`: This recipe performs dark data reduction
- `eris_ifu_detlin`: Detector’s linearity & non linear bad pixels
- `eris_ifu_distortion`: This recipe performs distortion correction
- `eris_ifu_flat`: This recipe performs flat field data reduction
- `eris_ifu_wavecal`: This recipe performs the wavelength calibration
- `eris_ifu_stdstar`: This recipe reconstructs data cubes from std stars
- `eris_ifu_jitter`: This recipe reconstructs data cubes from object
- `eris_ifu_combine_hdr1`: Combine science cubes using hdr1.
- `eris_ifu_combine`: This recipe combines jittered cubes into a single cube
- `eris_ifu_pupil`: This recipe monitors the telescope pupil position

7.2 Running the ERIS Pipeline Recipes

7.2.1 Getting Started with EsoRex

`EsoRex` is a command-line tool which can be used to execute the recipes of all standard VLT/VLTI instrument pipelines. With `EsoRex` in your path, the general structure of an `EsoRex` command line is

```
1> esorex [esorex options] [recipe [recipe options] [sof [sof]...]]
```

where options appearing before the recipe name are options for `EsoRex` itself, and options given after the recipe name are options which affect the recipe.

All available `EsoRex` options can be listed with the command

```
1> esorex --help
```

and the full list of available parameters of a specific recipe can be obtained with the command

```
1> esorex --help <recipe name>
```

The output of this command shows as parameter values the current setting, i.e. all modifications from a configuration file or the command line are already applied.

The listing of all recipes known to `EsoRex` can be obtained with the command

```
1> esorex --recipes
```
The last arguments of an EsoRex command are the so-called set-of-frames. A set-of-frames is a simple text file which contains a list of input data files for the recipe. Each input file is followed by an unique identifier (frame classification or frame tag), indicating the contents of this file. The input files have to be given as an absolute path, however EsoRex allows the use of environment variables so that a common directory prefix can be abbreviated. Individual lines may be commented out by putting the hash character (#) in the first column. An example of a set-of-frames is shown in the following:

```
$RAW_DATA/ERIS.2022-03-29T09:48:53.153.fits DARK
$RAW_DATA/ERIS.2022-03-29T09:50:36.645.fits DARK
$RAW_DATA/ERIS.2022-03-29T09:52:16.513.fits DARK
$RAW_DATA/ERIS.2022-03-29T09:53:47.996.fits DARK
#$RAW_DATA/ERIS.2022-03-29T09:55:04.515.fits DARK
```

These set-of-frames files will have to be created by the user using a text editor, for instance. Which classification has to be used with which recipe will be shown in section 7.3.

Finally, if more than one set-of-frames is given on the command-line EsoRex concatenates them into a single set-of-frames.

### 7.3 Data Organization

The ERIS-SPIFFIER pipeline recipes accept the following input tags and generates frames with the following output tags:

- **Recipe eris_ifu_dark**

  **Input files:**

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARK</td>
<td>Data</td>
<td>YES</td>
</tr>
</tbody>
</table>

  **Output files:**

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER_DARK_IFU</td>
<td>master dark</td>
</tr>
<tr>
<td>BPM_DARK</td>
<td>hot pixel map</td>
</tr>
</tbody>
</table>

- **Recipe eris_ifu_detlin**

  **Input files:**

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
</table>
### Output files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINEARITY_LAMP</td>
<td>Data</td>
<td>YES</td>
</tr>
</tbody>
</table>

### Input files:  

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARK_NS</td>
<td>raw data (dark)</td>
<td>YES</td>
</tr>
<tr>
<td>FIBRE_NS</td>
<td>raw data (ifu slices)</td>
<td>YES</td>
</tr>
<tr>
<td>FLAT_NS</td>
<td>raw data (flat)</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_NS</td>
<td>raw data (arcs)</td>
<td>YES</td>
</tr>
<tr>
<td>FIRST_WAVE_FIT</td>
<td>static (guess sol, wave, pos)</td>
<td>YES</td>
</tr>
<tr>
<td>REF_LINE_ARC</td>
<td>static (ref arc lines)</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_SETUP</td>
<td>static (guess sol, range fit)</td>
<td>YES</td>
</tr>
</tbody>
</table>

### Output files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTORTION</td>
<td>table with distortion correction coefficients</td>
</tr>
<tr>
<td>BPM_DIST</td>
<td>bad pixel map</td>
</tr>
<tr>
<td>SLITLET_POS</td>
<td>slitlets start/end positions</td>
</tr>
</tbody>
</table>

### Recipe eris_ifu_distortion

#### Input files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARK_NS</td>
<td>raw data (dark)</td>
<td>YES</td>
</tr>
<tr>
<td>FIBRE_NS</td>
<td>raw data (ifu slices)</td>
<td>YES</td>
</tr>
<tr>
<td>FLAT_NS</td>
<td>raw data (flat)</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_NS</td>
<td>raw data (arcs)</td>
<td>YES</td>
</tr>
<tr>
<td>FIRST_WAVE_FIT</td>
<td>static (guess sol, wave, pos)</td>
<td>YES</td>
</tr>
<tr>
<td>REF_LINE_ARC</td>
<td>static (ref arc lines)</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_SETUP</td>
<td>static (guess sol, range fit)</td>
<td>YES</td>
</tr>
</tbody>
</table>

### Recipe eris_ifu_flat

#### Input files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAT_LAMP</td>
<td>raw data (flat)</td>
<td>YES</td>
</tr>
<tr>
<td>BPM_DARK</td>
<td>chain (eris_ifu_dark)</td>
<td>YES</td>
</tr>
<tr>
<td>BPM_DETLIN</td>
<td>chain (eris_ifu_detlin)</td>
<td>RECOMMENDED</td>
</tr>
<tr>
<td>BPM_DIST</td>
<td>chain (eris_ifu_distortion)</td>
<td>RECOMMENDED</td>
</tr>
</tbody>
</table>

#### Output files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
BPM_FLAT
MASTER_FLAT

- Recipe eris_ifu_wavecal

Input files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAVE_LAMP</td>
<td>raw data (arc on)</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_LAMP</td>
<td>raw data (arc off)</td>
<td>YES</td>
</tr>
<tr>
<td>DISTORTION</td>
<td>chain (eris_ifu_distortion)</td>
<td>YES</td>
</tr>
<tr>
<td>MASTER_FLAT</td>
<td>chain (eris_ifu_mflat)</td>
<td>RECOMMENDED</td>
</tr>
<tr>
<td>FIRST_WAVE_FIT</td>
<td>static (guess sol, wave, pos)</td>
<td>YES</td>
</tr>
<tr>
<td>REF_LINE_ARC</td>
<td>static (ref arc lines)</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_SETUP</td>
<td>static (guess sol, range fit)</td>
<td>YES</td>
</tr>
</tbody>
</table>

Output files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAVE_LAMP_STACKED</td>
<td>stacked arc lamp frame</td>
</tr>
<tr>
<td>WAVE_LAMP_RESAMPLED</td>
<td>resampled arc lamp frame (qc)</td>
</tr>
<tr>
<td>WAVE_MAP</td>
<td>wavelength map</td>
</tr>
</tbody>
</table>

- Recipe eris_ifu_stdstar. This recipe support three cases that are different for DPR.TYPE value and corresponding input frame tag, to easy operations. The standard stars can be photometric standards (DPR.TYPE=STD,FLUX) used to determine the instrument efficiency and response on a limited set of stars observed in photometric nights; PSF standards (DPR.TYPE=PSF-CALIBRATOR), observed to compute the Strehl; other spectrophotometric stars and telluric standards (DPR.TYPE=STD). The corresponding set of frames differ on the raw frame tag and some static input frames, and are listed below:

  - with photometric std star as input:

Input files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD_FLUX</td>
<td>raw data (flux std)</td>
<td>YES</td>
</tr>
<tr>
<td>SKY_STD_FLUX</td>
<td>raw data (sky for flux std)</td>
<td>YES</td>
</tr>
<tr>
<td>DISTORTION</td>
<td>chain (eris_ifu_distortion)</td>
<td>YES</td>
</tr>
<tr>
<td>SLITLET_POS</td>
<td>chain (eris_ifu_distortion)</td>
<td>YES</td>
</tr>
<tr>
<td>MASTER_FLAT</td>
<td>chain (eris_ifu_mflat)</td>
<td>YES</td>
</tr>
<tr>
<td>BPM_FLAT</td>
<td>bad pixel map</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_MAP</td>
<td>wavelength map</td>
<td>YES</td>
</tr>
<tr>
<td>EXTCOEFF_TABLE</td>
<td>static (atm ext table)</td>
<td>YES</td>
</tr>
<tr>
<td>OH_SPEC</td>
<td>table with OH lines to get OH based wavelength solution</td>
<td>YES</td>
</tr>
<tr>
<td>RESPONSE_WINDOWS</td>
<td>static windows to compute resp.</td>
<td>YES</td>
</tr>
<tr>
<td>RESP_FIT_POINTS_CATALOG</td>
<td>static points to fit response</td>
<td>YES</td>
</tr>
<tr>
<td>TELL_MOD_CATALOG</td>
<td>static telluric model catalog</td>
<td>YES</td>
</tr>
</tbody>
</table>
FIT AREAS        static fit areas YES
QUALITY AREAS    static quality areas YES
FLUX_STD_CATALOG static flux std catalog YES
EFFICIENCY_WINDOWS static efficiency windows YES

Output files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKY_CUBE</td>
<td>sky obs with std star data cube</td>
<td></td>
</tr>
<tr>
<td>STD_FLUX_CUBE</td>
<td>flux std star data cube</td>
<td></td>
</tr>
<tr>
<td>STD_FLUX_CUBE_COADD</td>
<td>flux std star data cube resampled</td>
<td></td>
</tr>
<tr>
<td>STD_CUBE_MEAN</td>
<td>mean of STD_FLUX_CUBE_COADD</td>
<td></td>
</tr>
<tr>
<td>STD_FLUX_CUBE_MEDIAN</td>
<td>median of STD_FLUX_CUBE_COADD</td>
<td></td>
</tr>
<tr>
<td>EXTRACTION_MASK</td>
<td>mask used for extraction</td>
<td></td>
</tr>
<tr>
<td>SPECTRUM</td>
<td>extracted spectrum (from STD_FLUX_CUBE_COADD)</td>
<td></td>
</tr>
<tr>
<td>RESPONSE</td>
<td>instrument response</td>
<td></td>
</tr>
<tr>
<td>SPECTRUM_FLUXCAL</td>
<td>flux calibrated response</td>
<td></td>
</tr>
<tr>
<td>STD_FLUX_CUBE_COADD_FLUXCAL</td>
<td>flux calibrated STD_FLUX_CUBE_COADD</td>
<td></td>
</tr>
<tr>
<td>EFFICIENCY</td>
<td>instrument efficiency</td>
<td></td>
</tr>
</tbody>
</table>

– with PSF std input:

Input files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
<th>Required:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSF_CALIBRATOR</td>
<td>raw data (PSF std)</td>
<td>YES</td>
</tr>
<tr>
<td>SKY_PSF_CALIBRATOR</td>
<td>raw data (sky for PSF std)</td>
<td>YES</td>
</tr>
<tr>
<td>DISTORTION</td>
<td>chain (eris_ifu_distortion)</td>
<td>YES</td>
</tr>
<tr>
<td>SLITLET_POS</td>
<td>chain (eris_ifu_distortion)</td>
<td>YES</td>
</tr>
<tr>
<td>MASTER_FLAT</td>
<td>chain (eris_ifu_mflat)</td>
<td>YES</td>
</tr>
<tr>
<td>RPM_FLAT</td>
<td>bad pixel map</td>
<td>YES</td>
</tr>
<tr>
<td>WAVE_MAP</td>
<td>wavelength map</td>
<td>YES</td>
</tr>
<tr>
<td>EXTCOEFF_TABLE</td>
<td>static (atm ext table)</td>
<td>YES</td>
</tr>
<tr>
<td>OH_SPEC</td>
<td>table with OH lines to get</td>
<td>YES</td>
</tr>
<tr>
<td>RESPONSE</td>
<td>instrument response</td>
<td>OPTIONAL</td>
</tr>
</tbody>
</table>

Output files:

<table>
<thead>
<tr>
<th>DO category:</th>
<th>Explanation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKY_CUBE</td>
<td>sky obs with std star data cube</td>
</tr>
<tr>
<td>PSF_CUBE</td>
<td>PSF std star data cube</td>
</tr>
<tr>
<td>PSF_CUBE_COADD</td>
<td>PSF std star data cube resampled</td>
</tr>
<tr>
<td>PSF_CUBE_MEAN</td>
<td>mean of PSF_CUBE_COADD</td>
</tr>
<tr>
<td>PSF_CUBE_MEDIAN</td>
<td>median of PSF_CUBE_COADD</td>
</tr>
<tr>
<td>EXTRACTION_MASK</td>
<td>mask used for extraction</td>
</tr>
<tr>
<td>SPECTRUM</td>
<td>extracted spectrum (from PSF_CUBE_COADD)</td>
</tr>
<tr>
<td>SPECTRUM_FLUXCAL</td>
<td>flux calibrated response (*)</td>
</tr>
<tr>
<td>PSF_CUBE_COADD_FLUXCAL</td>
<td>flux calibrated PSF_CUBE_COADD (*)</td>
</tr>
</tbody>
</table>

(*) If RESPONSE is provided

– with normal std input:

Input files:
### DO category: STD
- **Explanation**: raw data (normal std)
- **Required**: YES

### DO category: SKY_STD
- **Explanation**: raw data (sky for normal std)
- **Required**: YES

### DO category: DISTORTION
- **Explanation**: chain (eris_ifu_distortion)
- **Required**: YES

### DO category: SLITLET_POS
- **Explanation**: chain (eris_ifu_distortion)
- **Required**: YES

### DO category: MASTER_FLAT
- **Explanation**: chain (eris_ifu_mflat)
- **Required**: YES

### DO category: BPM_FLAT
- **Explanation**: bad pixel map
- **Required**: YES

### DO category: WAVE_MAP
- **Explanation**: wavelength map
- **Required**: YES

### DO category: EXTCOEFF_TABLE
- **Explanation**: static (atm ext table)
- **Required**: YES

### DO category: OH_SPEC
- **Explanation**: table with OH lines to get
- **Required**: YES

### DO category: RESPONSE
- **Explanation**: instrument response
- **Required**: OPTIONAL

### Output files:

### DO category: SKY_CUBE
- **Explanation**: sky obs with std star data cube

### DO category: STD_CUBE
- **Explanation**: std star data cube

### DO category: STD_CUBE_COADD
- **Explanation**: std star data cube resampled

### DO category: STD_CUBE_MEAN
- **Explanation**: mean of STD_CUBE_COADD

### DO category: STD_FLUX_CUBE_MEDIAN
- **Explanation**: median of STD_CUBE_COADD

### DO category: EXTRACTION_MASK
- **Explanation**: mask used for extraction

### DO category: SPECTRUM
- **Explanation**: extracted spectrum (from STD_CUBE_COADD)

### DO category: SPECTRUM_FLUXCAL
- **Explanation**: flux calibrated response (*)

(*) If RESPONSE is provided

- Recipe eris_ifu_jitter (science object input)

### Input files:

### DO category: OBJ
- **Explanation**: raw data (object)
- **Required**: YES

### DO category: SKY_OBJ
- **Explanation**: raw data (sky for object)
- **Required**: YES

### DO category: DISTORTION
- **Explanation**: chain (eris_ifu_distortion)
- **Required**: YES

### DO category: SLITLET_POS
- **Explanation**: chain (eris_ifu_distortion)
- **Required**: YES

### DO category: MASTER_FLAT
- **Explanation**: chain (eris_ifu_mflat)
- **Required**: YES

### DO category: BPM_FLAT
- **Explanation**: bad pixel map
- **Required**: YES

### DO category: WAVE_MAP
- **Explanation**: wavelength map
- **Required**: YES

### DO category: EXTCOEFF_TABLE
- **Explanation**: static (atm ext table)
- **Required**: YES

### DO category: OH_SPEC
- **Explanation**: table with OH lines to get
- **Required**: YES

### DO category: RESPONSE
- **Explanation**: instrument response
- **Required**: OPTIONAL

### Output files:
<table>
<thead>
<tr>
<th>SKY_CUBE</th>
<th>sky obs with object data cube</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ_CUBE</td>
<td>std star data cube</td>
</tr>
<tr>
<td>OBJ_CUBE_COADD</td>
<td>std star data cube resampled</td>
</tr>
<tr>
<td>OBJ_CUBE_MEAN</td>
<td>mean of OBJ_CUBE_COADD</td>
</tr>
<tr>
<td>OBJ_CUBE_MEDIAN</td>
<td>median of OBJ_CUBE_COADD</td>
</tr>
<tr>
<td>EXTRACTION_MASK</td>
<td>mask used for extraction</td>
</tr>
<tr>
<td>SPECTRUM</td>
<td>extracted spectrum (from OBJ_CUBE_COADD)</td>
</tr>
<tr>
<td>SPECTRUM_FLUXCAL</td>
<td>flux calibrated response (*)</td>
</tr>
<tr>
<td>OBJ_CUBE_COADD_FLUXCAL</td>
<td>flux calibrated OBJ_CUBE_COADD (*)</td>
</tr>
</tbody>
</table>

(*) If RESPONSE is provided

### 7.4 Example of data reduction using EsoRex

A simple, typical data reduction procedure is described here.

We suggest the user to organize his data per type, observed band and camera setting. Dark frames may be grouped per detector DIT, frames to compute distortion and frames to compute detector non linearities may be organized per observed band. The detector DIT is given by the value of the FITS keyword DET SEQ1 DIT. The observed band is indicated by the value of the FITS keyword INS3 SPGW ID. The pre-optic setting is indicated by the value of INS3 SPXW ID. In the examples below we suppose the user has data acquired in band K_low and with the 100 mas pre-optic setting, and DIT=600. In the following examples /path_raw/ indicates the full path to the source tree directory containing raw data. All calibration frames have DPR.CATG='CALIB'. Science observations have DPR.CATG='SCIENCE'.

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

```
/path_raw/DARK/600/ERIS.2022-08-23T09:36:12.316.fits DARK
/path_raw/DARK/600/ERIS.2022-08-23T09:51:59.824.fits DARK
/path_raw/DARK/600/ERIS.2022-08-23T10:07:40.760.fits DARK
```

**Detector linearity flat field frames:** those calibration frames are characterized by DPR.TYPE='LINEARITY,LAMP'

```
/path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:09:50.882.fits LINEARITY_LAMP
/path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:10:07.455.fits LINEARITY_LAMP
/path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:10:23.047.fits LINEARITY_LAMP
/path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:10:38.240.fits LINEARITY_LAMP
```

1. We omit here the prefix HIERARCH ESO
Fibre frames, flat frames and arc lamp frames to compute distortions: those calibration frames have DPR.TYPE respectively equal to 'NS,DARK', 'NS,SLIT', 'NS,FLAT,DARK' 'NS,FLAT,FLAT' (usually 2 frames, one with calibration flat lamp switched off, one with the lamp switched on), 'NS,WAVE,DARK', 'NS,WAVE,LAMP'.

/path_raw/DISTORTION/K_low/ERIS.2020-03-14T11:46:25.168.fits DARK_NS
/path_raw/DISTORTION/K_low/ERIS.2020-03-14T11:46:53.132.fits FIBRE_NS
/path_raw/DISTORTION/K_low/ERIS.2020-03-14T11:53:22.863.fits FLAT_NS
/path_raw/DISTORTION/K_low/ERIS.2020-03-14T11:53:45.397.fits FLAT_NS
/path_raw/DISTORTION/K_low/ERIS.2020-03-14T11:54:08.861.fits WAVE_NS
/path_raw/DISTORTION/K_low/ERIS.2020-03-14T11:55:10.220.fits WAVE_NS

Standard flat field frames: those calibration frames are caracterized by DPR.TYPE='FLAT,DARK' and DPR.TYPE='FLAT,LAMP'

/path_raw/FLAT/K_low/100/ERIS.2020-02-28T16:27:43.232.fits FLAT_LAMP
/path_raw/FLAT/K_low/100/ERIS.2020-02-28T16:28:05.846.fits FLAT_LAMP ...
/path_raw/FLAT/K_low/100/ERIS.2020-02-28T16:28:45.566.fits FLAT_LAMP
/path_raw/FLAT/K_low/100/ERIS.2020-02-28T16:29:08.165.fits FLAT_LAMP

Arc lamp frames: those calibration frames are caracterized by DPR.TYPE='WAVE,DARK' and DPR.TYPE='WAVE,LAMP'

/path_raw/WAVE/K_low/100/ERIS.2022-02-28T17:55:44.753.fits WAVE_LAMP
/path_raw/WAVE/K_low/100/ERIS.2022-02-28T18:00:15.875.fits WAVE_LAMP

To determine the instrument efficiency and response the user may want to reduce also flux standard star frames: those calibration frames are caracterized by DPR.TYPE='STD,FLUX' or DPR.TYPE='SKY,STD,FLUX'

/path_raw/STD/K_low/100/ERIS.2022-02-27T09:17:05.775.fits STD_FLUX
/path_raw/STD/K_low/100/ERIS.2022-02-27T09:18:10.566.fits SKY_STD_FLUX

Strehl is computed on PSF standard star frames: those frames are caracterized by DPR.TYPE='PSF-CALIBRATOR' or DPR.TYPE='SKY,PSF-CALIBRATOR'

/path_raw/PSF/K_low/100/ERIS.2022-08-23T07:35:16.597.fits PSF_CALIBRATOR
/path_raw/PSF/K_low/100/ERIS.2022-08-23T07:36:55.413.fits SKY_PSF_CALIBRATOR

Additional standard frames are observed. Those frames are caracterized by DPR.TYPE='STD' or DPR.TYPE='SKY,STD'

/path_raw/STD/K_low/100/ERIS.2022-02-27T09:17:05.775.fits STD
/path_raw/STD/K_low/100/ERIS.2022-02-27T09:18:10.566.fits SKY_STD
science frames: those frames are characterized by DPR.CATG='SCIENCE', DPR.TYPE='OBJECT' or DPR.TYPE='SKY'

/path_raw/SCI/K_low/100/ERIS.2022-02-27T04:30:22.175.fits OBJ
/path_raw/SCI/K_low/100/ERIS.2022-02-27T04:30:54.620.fits SKY_OBJ

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 extention .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, preoptic and band, and to remove the other products after each recipe execution.

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

   /path_raw/DARK/600/ERIS.2022-08-23T09:36:12.316.fits DARK
   /path_raw/DARK/600/ERIS.2022-08-23T09:51:59.824.fits DARK
   /path_raw/DARK/600/ERIS.2022-08-23T10:07:40.760.fits DARK

   Then the user can generate the master dark with the command

   esorex eris_ifu_dark dark_sof

   This command will generate two files: eris_ifu_dark.bpm.fits (PRO.CATG=BPM_DARK), a hot pixel map, and eris_ifu_dark_master_dark.fits (PRO.CATG=MASTER_DARK_IFU), a master dark. For convenience we indicate with /path_cdb the full path to a directory containing relevant data reduction products.

   mv eris_ifu_dark_bpm.fits /path_pro/BPM_DARK.fits
   mv eris_ifu_dark_master_dark.fits /path_pro/MASTER_DARK_IFU_600.fits
   rm -rf out*.fits *.log

2. Then the user may generate a map of non linear pixels. A set of linearity raw flat field frames may be put in the ASCII file linearity_sof.

   /path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:09:50.882.fits LINEARITY_LAMP
   /path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:10:07.455.fits LINEARITY_LAMP
   /path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:10:23.047.fits LINEARITY_LAMP
   /path_raw/LINEARITY/K_low/ERIS.2022-02-26T20:10:38.240.fits LINEARITY_LAMP
   ...

   By default installation in the EsoRex configuration file ($HOME/.esorex/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.
The user can generate a non linearity bad pixel map (PRO.CATG=BPM_DETLIN) with the command:

```
esorex eris_ifu_detlin linearity_sof
```

This command will generate several files, including the non linearity bad pixel map, stored in the file "eris_ifu_detlin_bpm.fits".

```
mv eris_ifu_detlin_bpm.fits /path_pro/BPM_DETLIN_K_low.fits
rm -rf out*fits *.log
```

3. Then the user may determine the optical distortions (PRO.CATG=DISTORTION) and the slitlets positions (PRO.CATG=SLITLETS_POS).

The user will select all the files containing the string DISTORTION in their DPR.TYPE. Using those files and the line reference table of the corresponding band (K) and a drs setup table will generate an ASCII file distortion_sof:

```
/path_raw/ERIS.2022-03-14T11:28:19.007.fits DARK_NS
/path_raw/ERIS.2022-03-14T11:28:43.781.fits FIBRE_NS
/path_raw/ERIS.2022-03-14T11:53:22.863.fits FLAT_NS
/path_raw/ERIS.2022-03-14T11:53:45.397.fits FLAT_NS
/path_raw/ERIS.2022-03-14T11:54:08.861.fits WAVE_NS
/path_raw/ERIS.2022-03-14T11:55:10.220.fits WAVE_NS
/path_cdb/eris_ifu_first_fit.fits FIRST_WAVE_FIT
/path_cdb/eris_ifu_ref_lines.fits REF_LINE_ARC
/path_cdb/eris_ifu_wave_setup.fits WAVE_SETUP
```

To successfully run the distortion recipe are needed: a NS,DARK (DARK_NS) and a NS,SLIT (FIBRE_NS) frame; two FLAT_NS frames, one with calibration flat lamp switched on, one with the lamp switched off; two WAVE_LAMP frames, one with calibration flat lamp switched on, one with the lamp switched off; and the three static frames, the reference line list of the appropriate band, with tag REF_LINE_ARC, a static table, with tag FIRST_WAVE_FIT, that contains foreach instrument setting, first guess solutions for wavelengths and their pixel positions of a few reference lines, and the static table, with tag WAVE_SETUP, that contains the range value to be used in the first fit of the wavelength calibration solution, where with /path_cdb we have indicated the full path to the directory containing calibration data (see also Sec. 10).

The command

```
esorex eris_ifu_distortion distortion_sof
```

Will generate several products. Between those also eris_ifu_distortion_distortion.fits, a table containing information on the polynomial distortion coefficients, and eris_ifu_distortion_slitlet_pos.fits, a table containing information on the slitlet edges positions.

```
mv eris_ifu_distortion_distortion.fits /path_pro/DISTORTION_K_low.fits
mv eris_ifu_distortion_slitlet_pos.fits /path_pro/SLITLETS_POS_K_low.fits
rm -rf out*fits *.log
```

4. Then one selects the raw flat fields and list them in an ASCII file flat_sof together with some static calibrations and previously obtained products:
5. Then one lists raw arc lamp frames and additional static calibration frames and previously obtained master calibrations in an ASCII file wcal_sof:

```
/path_raw/WAVE/ERIS.2022-02-28T17:55:44.753.fits WAVE_LAMP
/path_raw/WAVE/ERIS.2022-02-28T18:00:15.875.fits WAVE_LAMP
/path_pro/MASTER_FLAT_K_100.fits MASTER_FLAT
/path_pro/DISTORTION_K_100.fits DISTORTION
/path_cdb/eris_ifu_ref_lines.fits REF_LINE_ARC
/path_cdb/eris_ifu_wave_setup.fits WAVE_SETUP
/path_cdb/eris_ifu_first_fit.fits FIRST_WAVE_FIT
```

The command:
```
esorex eris_ifu_wavecal wcal_sof
```
generates several frames, including the master wavelength map, stored in the file “eris_ifu_wave_map.fits”.

```
mv eris_ifu_wave_map.fits /path_pro/WAVE_MAP_K_100.fits
mv eris_ifu_bpm.fits /path_pro/BPM_FLAT_K_100.fits
rm -rf out*fits *.log
```

6. To create the instrument response the user will reduce the flux standard star data:

```
/path_raw/OBJNOD/ERIS.2022-02-26T06:09:19.358.fits STD_FLUX
/path_raw/OBJNOD/ERIS.2022-02-26T06:10:10.996.fits SKY_STD_FLUX
/path_pro/SLITLET_POS_K_100.fits SLITLET_POS
/path_pro/DISTORTION_K_100.fits DISTORTION
/path_pro/BPM_FLAT_K_100.fits BPM_FLAT
```
The command:

```bash
esorex eris_ifu_stdstar stdstar_sof
```

generates several frames, including the instrument RESPONSE frame.

```bash
mv eris_ifu_stdstar_response.fits /path_pro/RESPONSE_K_100.fits
rm -rf out*fits *.log
```

7. Finally the user will reduce the science data:

```bash
/path_raw/OBJNOD/ERIS.2022-02-27T06:09:19.358.fits OBJ
/path_raw/OBJNOD/ERIS.2022-02-27T06:10:10.996.fits SKY_OBJ
/path_raw/OBJNOD/ERIS.2022-02-27T06:11:33.949.fits OBJ
/path_raw/OBJNOD/ERIS.2022-02-27T06:13:08.204.fits SKY_OBJ
/path_pro/SLITLET_POS_K_100.fits SLITLET_POS
/path_pro/DISTORTION_K_100.fits DISTORTION
/path_pro/BPM_FLAT_K_100.fits BPM_FLAT
/path_pro/MASTER_FLAT_K_100.fits MASTER_FLAT
/path_pro/WAVE_MAP_K_100.fits WAVE_MAP
/path_pro/RESPONSE_K_100.fits RESPONSE
/path_ref/OH_SPEC.fits OH_SPEC
/path_ref/EXTCOEFF_TABLE.fits EXTCOEFF_TABLE
```

The command:

```bash
esorex eris_ifu_jitter sci_sof
```

generates several frames, including a reconstructed cube of each target observation, stored in the file “eris_ifu_jitter_obj_cube_00#.fits” (#=0,1,…,N), sky frames, stored in the file “eris_ifu_jitter_sky_cube_00#.fits” (#=0,1,…,N), a mosaic cube coadding each single cube observation, stored in the file “eris_ifu_jitter_obj_cube_coadd.fits” its median and mean images, stored in the file “eris_ifu_jitter_obj_cube_median.fits” and “eris_ifu_jitter_obj_cube_mean.fits”.
8 Known Issues

8.1 Use ESO Reflex

We recommend to reduce the data with ESOReflex. This release provides a workflow for ERIS-SPIFFIER data reduction (eris_spiffier.xml). The reconstructed cube resampling is RAM intensive, and requires some minutes. We could reduce most of commissioning and Science Verification data with 20 GB of RAM but we recommend users to have at least 32 GB of RAM. In a few cases, if observations imply large offsets over the FOV, the user may require more RAM.

8.2 Combining Multiple Cubes (possibly observed with different OBs)

User may have observations of the same object acquired by different OBs or template start/end sequences. They may want to combine them. Alternatively users may want not to combine some cubes but only the others into the final 3D cube mosaic. This release provides a recipe, eris_ifu_combine_hdr1, that allows to do this. The user has to do the following steps:

1. Run the eris_ifu_jitter recipe to reconstruct the individual cubes. The user may set the parameter cube.combine=FALSE to skip the combine step.

2. Prepare the combine.sof, that may look like the following:

   eris_ifu_jitter_obj_cube_000.fits OBJECT_CUBE
   eris_ifu_jitter_obj_cube_001.fits OBJECT_CUBE
   eris_ifu_jitter_obj_cube_002.fits OBJECT_CUBE
   eris_ifu_jitter_obj_cube_003.fits OBJECT_CUBE

   Run the eris_ifu_combine_hdr1 recipe:
   esorex eris_ifu_combine_hdr1 combine.sof

3. By default, the WCS keywords in each object cube will be used to resample the data onto the same grid. In case the WCS keywords do not align the different cubes well, users can generate an ASCII file (default name offset.list) containing the corresponding offsets of each cube, for example like (here in pixel units)

   0 0
   -10 0
   0 0
   0 10

   , and run the eris_ifu_combine_hdr1 recipe:
   esorex eris_ifu_combine_hdr1 --offset_mode=FALSE --name_i=offset.list combine.sof

   This release includes also the recipe eris_ifu_combine. This combines cubes using an algorithm similar to what was implemented for the SINFONI pipeline. Differently from eris_ifu_combine_hdr1, this recipe does not resample each data cube before combining them, which results in a much faster cube combination even if the comination methods are less than the other. The eris_ifu_combine recipe provides the ability to apply an initial percentile clipping and a consecutive kappa-sigma-clipping (see parameters pclip, ks_clip, kappa). For more details see Section 12.10).
8.3 Correcting possible residual minor shifts of the observed object

The observed object raw frame may have a small shift along the cross-dispersion direction, typically within the range [-2, 2] pixels, depending on telescope pointing position and the rotation angle of the instrument. This may affect the quality of the reconstructed frame, where the spectrum wraps around from one side of the cube to the other. To correct this effect, the user may use a dedicated parameter, `derot_corr`, and set it to the small shift. A positive number will shift the science cube to the right hand side, and the column(s) on the right edge will wrap to the left edge; a negative number will shift to the left hand side, and the left edge will be wrap to the right. With current release this correction is automatically done by the pipeline as the parameter `derot_corr` default is 'auto', that means the pipeline automatically estimates (from a model based on sampled data) the correct value of this parameter. If the user is not happy changing `derot_corr` parameter value overwrites the automatic setting.

8.4 Combining Multiple Cubes

During instrument commissioning we found rare cases where objects with very different WCS were observed in the same template sequence. This may lead to a combined cube of large size that may require more RAM of what the users has. In order to prevent the process is killed by the Linux Kernel we decided to define a parameter `--max-cubes-centres-dist` that sets the maximum allowed distance between the reconstructed cubes, for which a combined cube is created. Current default is set to 240 pixels. This corresponds to a maximum RAM usage of approximatively 20 GB. If the (estimated) computed maximum distance between reconstructed cubes is greater than that value, the pipeline does not generate the `OBJECT_CUBE_COADD` data product, nor the extracted spectrum (that is computed on the `OBJECT_CUBE_COADD`). If the user has enough RAM and would like to obtain the missing products, she/he may change the `max-cubes-centres-dist` parameter value.

To (slightly) reduce the cube resampling computation time the user may reduce the value of the parameter `method.loop-distance` to 1. Often this does not worsen the reconstructed cube quality (see Figure 8.1). The default `method.loop-distance=3` was set to optimise quality over a large data set even if this value implies a longer computation.

As the `eris_ifu_jitter` recipe spend a significant amount of data reduction time during the combination of the data cubes, that is an HDRL based parallelised data reduction step, it is obviously important to be able to run the recipe with a multi-core CPU. For example the reduction of an observation of RMC on a 100mas setting, that generates a combined cube of 94x94x2147 pixels, passed from 2:02.04 (mm:ss) on a simple Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz with 4 physical cores (8 threads), to 1:22.65 (mm:ss) using a 12th Gen Intel(R) Core(TM) i9-12900K with 16 cores (8+8) and 24 threads (16+8).

If the user does not want to create the combined cube, one may set the parameter `cube.combine` to false. In this case also the extracted spectrum will not be created.

8.5 Refining the Wavelength Calibration Using Sky Lines

The static frame with PRO.CATG OH_SPEC is now indicated as required input for the recipes `eris_ifu_stdstar` and `eris_ifu_jitter`. In the current pipeline release it is actually a recommended input. If the user does not input that frame the `eris_ifu_stdstar` and `eris_ifu_jitter` may still successfully reduce data,
Figure 8.1: This figure displays results obtained for the same object (V V2423 Ori) observation. The cube quality is almost identical thus we show only one plane of a datacube obtained with parameter defaults. The quality is good. With an Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz, the computations times are respectively of 104.22 s, 109.18 s, and 120.15 s. This means that the user may slightly improve computation time by reducing the value of the method.loop-distance. As the speed-up improvement is only of 15 %, we recommend to keep parameters defaults.

but their wavelength calibration may be less accurate, as it will be based only on the solution obtained from the daily calibrations provided by the lamp wavelength flat. This calibration can be affected by instrumental flexures (SPIFFIER is mounted on the Cassegrain focus of UT4). By providing the OH_SPEC fits table the wavelength solution initially computed from the lamp is refined using [OH] lines in the spectrum.

8.6 Missing BPM_DARK in Reflex DataOrganiser

To simplify ERIS-SPIFFIER operations we associate as input BPM_DARK of the flat recipe one with a DIT of 10s and NDIT=1. This is a dark acquired every day for QC. The OCA rules of CalSelector have been updated accordingly. Should the user have data acquired before, it might be that the ESOReflex data set chooser is greyed-out, indicating Missing BPM_DARK data. To solve this problem the user has to add DARKS of DIT 10s and NDIT=1 to the input data.

8.7 Wiggles Visible in Under-Sampled Data Cubes

If IFU data is spatially under-sampled, ripples can appear in the extracted spectrum from the resampled cube. This is an unavoidable interpolation effect. A example of this is apparent in a standard star exposure taken in the 100mas scale with AO and in the K-middle passband (see figure 8.2). The standard star is spatially under-sampled and the majority of its flux falls in a single rectangular pixel. If the spectrum is extracted over a single
pixel aperture the spectrum will have very apparent undulations. This is a known phenomena and has been described for KMOS in [?] (in particular, see their figures 12 and 13). However, if one integrates the flux over a reasonable aperture the wiggles go away, as the total flux is preserved (see figure 8.3).

**Figure 8.2:** An example of extracted spectrum ‘wiggles’ in an under-sampled standard star. To emphasise the undersampling, the median collapsed cube has been scaled to the image minimum and maximum flux levels (left panel). Clearly, the majority of the standard star’s flux falls into a single rectangular pixel. When extracting through the data cube over an aperture close to this single pixel, the undulating spectrum becomes apparent (right panel).
Figure 8.3: The identical under-sampled standard star of figure 8.2, but differently scaled (left panel). Clearly, the majority of the standard star’s flux falls into a single rectangular pixel. When extracting through the data cube over a reasonable aperture the wiggles disappear (right panel).
8.8 Flux calibrated spectrum defined on a wavelength range shorter than the one of the extracted spectrum

This problem is due to the spectral response, that maybe defined on a wavelength range shorter than the one of the extracted spectrum, thus when it is applied to it to do the flux calibration, it may reduce its range (because the division between spectra is controlled by HDRL, that does not compute points in range where one of the two operands is not defined). This effect may occur in particular in H band. We show in Figure 8.4 an example, obtained with the demo data, for LTT3218 in H band. The figure shows a comparison between the spectral response (smoothed) in red, and the raw response in blue, evidencing the fit points used (green dots). The spectral response is defined only over the range where are valid fit points.

The fit points are considered valid when they do not fall in atmospheric high absorption regions (where are present strong telluric lines). The latter are defined by the static input with PRO.CATG HIGH_ABS_REGIONS. As we are in H band and there are two regions respectively in the interval [1.3.7,1.506] $\mu$m and [1.736,1.986] $\mu$m even if the static input with PRO.CATG RESP_FIT_POINTS_CATALOG, has for LTT3218, in those regions a few points defined, those are not considered as clipped by the regions defined in HIGH_ABS_REGIONS.

A user who would like to extend the definition of the response should edit the static input with PRO.CATG HIGH_ABS_REGIONS corresponding to the H band, and shorten the interval of the high absorption regions. Additionally maybe required to add fit points in the fits file with PRO.CATG RESP_FIT_POINTS_CATALOG, in the extention corresponding to the observed flux standard star, in our case LTT3218, in extension 5. However we warn that this may imply that as consequence of those changes, the response, that is determined with a spline cubic fit to the valid fit points, may have sharp variations, due to the fact the fit of some line present in the trimmed high absorption region.

8.9 Possible failure of recipe eris_ifu_jitter in case –crea-phase3=true and –sky_tweak=1

Occasionally the recipe eris_ifu_jitter may fail if –crea-phase3=true and –sky_tweak=1. This because the sky_tweak modifies the sky background level and this affects the computation of a FITS keyword added as part of the phase3 data format. We are looking into a possible fix. In the meantime the user has two options:

- Deactivate creation of phase3 data format products (which is used internally by ESO to add FITS keyword
information required for the injection in the phase3 database to add metadata used later for archive catalog frame classification/search/association). This solves the problem.

- If the user does not require to apply the Davies method for sky modeling and residual correction, an option is also to set sky_tweak to 0. This usually solves the problem.

### 8.10 Obtaining proper flux calibrated results

To obtain proper flux calibrated results the response must be accurate. As described in section 13.2.11 the response is proportional to the ratio between the flux of the observed standard star provided by a reference catalog and the total flux of the standard star obtained extracting the SPIFFIER data. It is important to be sure to have integrated all flux of the observed standard star. Thus the extraction \texttt{radius} must be large enough to include all flux of the star (usually four times the start PSF sigma is a good value), but not too large to introduce noise from the adjacent SKY. The flux std stars are all observed in 250 mas setting without AO. Usually for most of them a value of 10 to 12 pixels is appropriate, depending on the seeing observing conditions.

### 8.11 Determination of response and spectrum flux calibration based on photometric standard stars not part of the calibration plan

The ERIS-SPIFFIER pipeline allows to flux calibrate combined data cubes and the corresponding extracted spectra, if the user determines first the instrument response by processing a spectro-photometric standard. The spectro photometric STD stars observed by ERIS are listed in the static input catalog with PRO.CATG \texttt{FLUX_STD_CATALOG}. Those are a few, but the ERIS calibration plan ensures they are observed in photometric conditions and with proper frequency. Users willing to flux calibrate the data with another STD star have to determine the response with their own data reduction, in the same format as the one created by the pipeline (at least for the first extension with the smoothed response) and then use it as input of the jitter recipe. We warn that the user may have to request to observe those stars and those maybe observed not in photometric conditions. It maybe possible to eventually determined the response automatically on another standard star as follows:

- The user needs to have the tabulated flux of the standard as observed by Space. This has to be in the same format as the fluxes contained in extensions 2-6 of the \texttt{FLUX_STD_CATALOG} table.

- Add an entry in the 1st table extension listing:
  - The \texttt{ext_id} integer number where the tabulated flux standard spectrum will be stored
  - The \texttt{name} of the standard star, as a string.
  - Its \texttt{ra} (right ascension), as a double precision number.
  - Its \texttt{dec} (declination), as a double precision number.

- Add the corresponding spectrum in the table extension corresponding to \texttt{ext_id}. The wavelength information has to go in a column named \texttt{LAMBDA}, the flux in a column named \texttt{FLUX}, the bin width (usually 1.0 for unbinned data) in a column named \texttt{BIN_WIDTH}.
• Upgrade the table `RESP_FIT_POINTS_CATALOG` in a similar manner. In the extension corresponding to the new std star are indicated the wavelengths at which the “raw” response determined by the pipeline should be fit. These have to be stored in a double precision column named 'LAMBDA'. Their position is refined by the standard star continuum, chosen not to fall in a region of telluric lines. As first approximation one may take the same values chosen for the other flux standard stars.

8.12 Possible failure of recipe eris_ifu_jitter in case –mask_method is set to 'fit'

The recipe `eris_ifu_jitter` may fail when –mask_method is set to 'fit' in case the image of a collapsed 3D data cube does not show an object with a good contrast with respect to the sky background. In this case the extraction with –mask_method set to 'fit' may fail as the 2D Gaussian fit does not converge. The user may eventually choose to set –mask_method to 'max' to get a spectrum, even if in such cases probably the extracted spectrum is not very meaningful. An example of such a case is shown in Figure 8.5.

![Figure 8.5](image)

Figure 8.5: Here we show the image of a 3D data cube mean collapsed along the wavelength axis. In this case the object cannot be distinguished from the background this leading to a failure of the extraction in case –mask_method is set to 'fit'. Using –mask_method=max one can still obtain a spectrum.
9 ERIS Data Description

In figure 9.1 we have represented how a possible observation target, (a), is imaged on the detector, (b). The 32 input source’s slices generated by the two image slicers are imaged on the detector. These are also called slitlets and appear as vertical strips. Each slitlet has two edges, which can be seen by looking at the horizontal traces of sky emission lines or atmospheric absorption from the hydroxyl molecule OH. Those lines also show that the slitlets are arranged in a brick-wall pattern, to ease the detection of their edges, and that the first slitlet has a left edge not coincident with the left border of the detector. This information, together with the actual position of each slitlet edge is necessary to reconstruct the input FOV image. The spatial information (both X and Y directions on the sky) is distributed along the horizontal direction alone (the X sky coordinate in each slitlet and the Y sky coordinate in the different slitlets) while the wavelength information is along the vertical direction, increasing downwards. The signal from a point-like object in the FOV with a finite PSF appears as a vertical stripe in each slitlet where its emission is relevant and it is superimposed on the sky background, more uniform along the spatial direction except in coincidence of emission lines visible as horizontal bright stripes. During an exposure the detector can be hit by cosmic rays.

Figure 9.1: A possible target for observation (a) and a typical raw science frame layout (b).
ERIS-SPIFFIER data reduction often involves pair of frames: calibrations in which an “on” frame is acquired with a calibration lamp switched on and an “off” frame recorded with the same lamp switched off, or observations of science objects and of the sky background. Additionally, dark frames are taken as well as special fibre flat frames used to compute optical distortions. Fibre flats are taken with the calibration fibre moved along the y-axis perpendicular to the image slices.

Each kind of raw frame is typically associated to a single ERIS-SPIFFIER pipeline recipe, i.e., the recipe assigned to the reduction of that specific frame type. In the pipeline environment this recipe would be executed automatically. The recipe to compute optical distortions takes as input several kinds of raw frames, including slit-fibre flats, and standard pairs of flat field and arc lamp frames.

In the following sections after a brief description of how most of the raw data look like, all raw ERIS-SPIFFIER data frames are listed, together with the keywords used for their classification and correct association. 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. To classify the SPIFFIER raw data normally are sufficient the values of three dedicated FITS keywords, “DPR TYPE”, “DPR CATG”, “DPR TECH” (and the values of INSTRUME and “SEQ ARM”). Other FITS keywords, used to specify the preoptics and grating settings and the DIT are used to associate data. Those are indicated in the following subsections.

9.1 Darks

These are frames obtained with the shutter closed, acquired to evaluate the detector dark current level, build a master frame, used for quality control and instrument health diagnostic.

Frame tag: DARK
Processed by: eris_ifu_mdark

<table>
<thead>
<tr>
<th>Classification Keywords</th>
<th>Association Keywords</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR.CATG = CALIB</td>
<td>DET.DIT</td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = DARK</td>
<td>TPL.START</td>
<td></td>
</tr>
<tr>
<td>DPR.TECH = IMAGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTRUME = ERIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ.ARM = SPIFFIER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An example is shown in Table 9.1.

9.2 Linearity frames

A ramp of lamp-on frames and lamp-off frames is acquired at increasing detector DITs to evaluate the detector pixel linearity, determine the gain, and compute a map of non linear pixels.

Frame tag: LINEARITY_LAMP
Processed by: eris_ifu_detlin
9.3 North-South test data

A set of frames, one obtained illuminating all slitlet edges with a slit, a dark, a flat and an arc lamp are acquired to determine the detector distortions.

Frame tags: DARK_NS, FIBRE_NS, FLAT_NS, WAVE_NS
Processed by: eris_ifu_distortion

An example is shown in Table 9.2.

9.4 Flats

A set of usually five flat on frames and five flat off frames are acquired to determine a master flat to correct for pixel to pixel detector non uniformities and larger scale variations.

Frame tags: FLAT_LAMP
Processed by: eris_ifu_mflat

An example is shown in Table 9.3.
9.5 Arc lamp frames

A pair of arc lamp on and off frames are acquired to perform the wavelength calibration.

**Frame tags:** WAVE_LAMP  
**Processed by:** eris_ifu_wavecal

<table>
<thead>
<tr>
<th>Classification Keywords</th>
<th>Association Keywords</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR.CATG = CALIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = WAVE, DARK</td>
<td>TPL.START</td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = WAVE, LAMP</td>
<td>INS3.SPXW.ID</td>
<td></td>
</tr>
<tr>
<td>DPR.TECH = IFU</td>
<td>INS3.SPXW.NAME</td>
<td></td>
</tr>
<tr>
<td>INSTRUME = ERIS</td>
<td>INS3.SPGW.ID</td>
<td></td>
</tr>
<tr>
<td>SEQ.ARM = SPIFFIER</td>
<td>INS3.SPGW.NAME</td>
<td></td>
</tr>
</tbody>
</table>

An example is shown in Table 9.5.

9.6 Flux standard star frames

A frame pair obtained observing a flux standard star and a sky frame are acquired to determine the instrument efficiency and the instrument response.

**Frame tags:** STD_FLUX  
**Processed by:** eris_ifu_stdstar

<table>
<thead>
<tr>
<th>Classification Keywords</th>
<th>Association Keywords</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR.CATG = CALIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = STD, FLUX</td>
<td>TPL.START</td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = SKY, STD, FLUX</td>
<td>INS3.SPXW.ID</td>
<td></td>
</tr>
<tr>
<td>DPR.TECH = IFU</td>
<td>INS3.SPXW.NAME</td>
<td></td>
</tr>
<tr>
<td>INSTRUME = ERIS</td>
<td>INS3.SPGW.ID</td>
<td></td>
</tr>
<tr>
<td>SEQ.ARM = SPIFFIER</td>
<td>INS3.SPGW.NAME</td>
<td></td>
</tr>
</tbody>
</table>

An example is shown in Table 9.6.

9.7 PSF standard star frames

A frame obtained observing a PSF standard star are acquired to determine the Strehl.

**Frame tags:** PSF_CALIBRATOR  
**Processed by:** eris_ifu_stdstar

<table>
<thead>
<tr>
<th>Classification Keywords</th>
<th>Association Keywords</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR.CATG = CALIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = PSF-CALIBRATOR</td>
<td>TPL.START</td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = SKY, PSF-CALIBRATOR</td>
<td>INS3.SPXW.ID</td>
<td></td>
</tr>
<tr>
<td>DPR.TECH = IFU</td>
<td>INS3.SPXW.NAME</td>
<td></td>
</tr>
<tr>
<td>INSTRUME = ERIS</td>
<td>INS3.SPGW.ID</td>
<td></td>
</tr>
<tr>
<td>SEQ.ARM = SPIFFIER</td>
<td>INS3.SPGW.NAME</td>
<td></td>
</tr>
</tbody>
</table>
An example is shown in Table 9.7.

### 9.8 Other standard star frames

Some observations of other standard stars can be performed.

**Frame tags:** STD

**Processed by:** eris_ifu_stdstar

<table>
<thead>
<tr>
<th>Classification Keywords</th>
<th>Association Keywords</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR.CATG = CALIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = STD</td>
<td>TPL.START</td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = SKY,STD</td>
<td>INS3.SPXW.ID</td>
<td></td>
</tr>
<tr>
<td>DPR.TECH = IFU</td>
<td>INS3.SPXW.NAME</td>
<td></td>
</tr>
<tr>
<td>INSTRUME = ERIS</td>
<td>INS3.SPGW.ID</td>
<td></td>
</tr>
<tr>
<td>SEQ.ARM = SPIFFIER</td>
<td>INS3.SPGW.NAME</td>
<td></td>
</tr>
</tbody>
</table>

An example is shown in Table 9.8.

### 9.9 Pupil monitoring frames

Some observations of other standard stars can be performed.

**Frame tags:** PUPIL_LAMP

**Processed by:** eris_ifu_pupil

<table>
<thead>
<tr>
<th>Classification Keywords</th>
<th>Association Keywords</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR.CATG = CALIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = PUPIL,LAMP</td>
<td>TPL.START</td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = SKY,PUPIL,LAMP</td>
<td>INS3.SPXW.ID</td>
<td></td>
</tr>
<tr>
<td>DPR.TECH = IFU</td>
<td>INS3.SPXW.NAME</td>
<td></td>
</tr>
<tr>
<td>INSTRUME = ERIS</td>
<td>INS3.SPGW.ID</td>
<td></td>
</tr>
<tr>
<td>SEQ.ARM = SPIFFIER</td>
<td>INS3.SPGW.NAME</td>
<td></td>
</tr>
</tbody>
</table>

### 9.10 Science observations

Science (and usually sky) frames are acquired to do science.

**Frame tags:** OBJ

**Processed by:** eris_ifu_jitter

<table>
<thead>
<tr>
<th>Classification Keywords</th>
<th>Association Keywords</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR.CATG = SCIENCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = OBJECT</td>
<td>TPL.START</td>
<td></td>
</tr>
<tr>
<td>DPR.TYPE = SKY</td>
<td>INS3.SPXW.ID</td>
<td></td>
</tr>
<tr>
<td>DPR.TECH = IFU</td>
<td>INS3.SPXW.NAME</td>
<td></td>
</tr>
<tr>
<td>INSTRUME = ERIS</td>
<td>INS3.SPGW.ID</td>
<td></td>
</tr>
<tr>
<td>SEQ.ARM = SPIFFIER</td>
<td>INS3.SPGW.NAME</td>
<td></td>
</tr>
</tbody>
</table>

An example is shown in Table 9.10.
Figure 9.2: (a) a raw dark frame; (b) a raw linearity frame; (c) a raw slit fibre flat frame to compute optical distortions; (d) a raw flat on frame; (e) a raw arc lamp on frame; (f) a raw std star frame.
10 Static Calibration Data

10.1 First Wave Fit

This static table contains foreach instrument setting a list of a few lines, their wavelength and approximate position on the detector, that are used to perform the first step of the wavelength calibration. In this way one is sure to have a robust initial wavelength calibration solution that is later refined using the several more weaker lines that can be found on the detector.

Frame Tag: FIRST_WAVE_FIT
Classification keywords: PRO.CATG = FIRST_WAVE_FIT

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instrument</td>
<td>string</td>
<td>value of INSTRUME</td>
</tr>
<tr>
<td>band</td>
<td>string</td>
<td>value of observation band</td>
</tr>
<tr>
<td>lamps</td>
<td>string</td>
<td>calibration lamp id</td>
</tr>
<tr>
<td>block</td>
<td>int</td>
<td>block id</td>
</tr>
<tr>
<td>position</td>
<td>int</td>
<td>position id</td>
</tr>
<tr>
<td>wavelength</td>
<td>double</td>
<td>Line wavelength um</td>
</tr>
<tr>
<td>Flux</td>
<td>float</td>
<td>Relative line flux</td>
</tr>
<tr>
<td>Ion</td>
<td>string</td>
<td>Ion from which the line originates</td>
</tr>
<tr>
<td>Quality</td>
<td>int</td>
<td>Quality flag (0: undetected line, 1: etc.)</td>
</tr>
<tr>
<td>Comment</td>
<td>string</td>
<td>Optional column for further remarks</td>
</tr>
</tbody>
</table>

An example is shown in Table 10.1.

10.2 Reference Line Catalogue

This is a static table containing all lines that are used in the final fit of the wavelength calibration.

Frame Tag: REF_LINE_ARC
Classification keywords: PRO.CATG = REF_LINE_ARC

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavelength</td>
<td>double</td>
<td>Line wavelength um</td>
</tr>
<tr>
<td>intensity</td>
<td>double</td>
<td>value of line intensity</td>
</tr>
<tr>
<td>ignored</td>
<td>int</td>
<td>if 0 the line is ignored</td>
</tr>
<tr>
<td>element</td>
<td>string</td>
<td>chemical line source</td>
</tr>
<tr>
<td>state</td>
<td>int</td>
<td>state of ionisation</td>
</tr>
</tbody>
</table>

An example is shown in Table 10.2.
10.3 Wavelength Setup Table

This is a static table and indicates for each instrument setting the range to be used in the fit of the few reference lines used during the first step of the wavelength calibration.

Frame Tag: WAVE_SETUP
Classification keywords: PRO.CATG = WAVE_SETUP, INS3.SPXW.ID = J_low, or H_Low, or K_low

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>band</td>
<td>string</td>
<td>observing band id</td>
</tr>
<tr>
<td>fwhm</td>
<td>double</td>
<td>Line FWHM pix</td>
</tr>
<tr>
<td>s_range</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>c_range</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>ri_dispersion</td>
<td>double</td>
<td>dispersion</td>
</tr>
<tr>
<td>ri_centrallambda</td>
<td>double</td>
<td>central wavelength</td>
</tr>
</tbody>
</table>

An example is shown in Table 10.3.

10.4 Catalog of Flux Standards Stars

Frame Tag: FLUX_STD_CATALOG
Classification keywords: PRO.CATG = FLUX_STD_CATALOG

This is a multi-extension table containing in the first extension information on the extension number (ext_id) where is located the spectrum of a star, its name (name), and values of right ascension (ra) and declination (dec). The other extensions contain the spectra of wavelength (LAMBDA), flux (FLUX) in physical units (erg/ph/cm²/s/A) and sampling bin width.

10.5 Atmospheric extinction table

Frame Tag: EXTCOEFF_TABLE
Classification keywords: PRO.CATG = EXTCOEFF_TABLE

This is a table containing information on the extinction (EXTINCTION) as a function of wavelength (LAMBDA). It is used to determine the instrument response and flux calibrate the data.

10.6 Efficiency Windows

Frame Tag: EFFICIENCY_WINDOWS
Classification keywords: PRO.CATG = EFFICIENCY_WINDOWS

This is a table indicating the start and end wavelength of each region where the efficiency is evaluated.
10.7 Fit Areas

Frame Tag: FIT.Areas
Classification keywords: PRO.CATG = FITAreas

This is a table indicating the start and end wavelength of each region where the a fit of the raw response spectrum is performed.

10.8 Fit Areas

Frame Tag: FIT.Areas
Classification keywords: PRO.CATG = FITAreas

This is a table indicating the start and end wavelength of each region where the a fit of the raw response spectrum is performed.

10.9 High Abs Regions

Frame Tag: HIGH_Abs_REGIONS
Classification keywords: PRO.CATG = HIGH_Abs_REGIONS

This is a table indicating the start and end wavelength of each region where the spectrum is dominated by high absorption telluric regions. Those regions are not used to evaluated the instrument response.

10.10 Quality Areas

Frame Tag: QUALITY_AREAS
Classification keywords: PRO.CATG = QUALITY_AREAS

This is a table indicating the start and end wavelength of each region where the quality of the match of a telluric model to the actual data is evaluated.

10.11 Response Windows

Frame Tag: RESPONSE_WINDOWS
Classification keywords: PRO.CATG = RESPONSE_WINDOWS

This is a table indicating the start and end wavelength of each region where the response fit is performed.
10.12 Catalog of fit points to fit the Response

Frame Tag: RESP_FIT_POINTS_CATALOG
Classification keywords: PRO.CATG = RESP_FIT_POINTS_CATALOG

This is a multi extension table with format similar to the FLUX_STD_CATALOG listing for each of the flux standards stars in the FLUX_STD_CATALOG, the wavelength of the points to be used to perform a fit.

10.13 Catalogue of Telluric models

Frame Tag: TELL_MOD_CATALOG
Classification keywords: PRO.CATG = TELL_MOD_CATALOG

This is a multi extension table with 73 spectra of telluric standards corresponding to different parameters of an atmospheric model. These are used during determination of the instrument response to identify the model that best matches to the data and then divide the observed spectrum by the best matching model to reduce the effect of the tellurics when evaluating the instrument response curve.
11 Data Reduction

11.1 The ERIS Data Reduction Pipeline

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

11.2 Data reduction overview

The principle of integral field spectroscopy is described by figure 11.1. A two-dimensional image of the sky is separated by an image slicer into several components. Those are then aligned on a slit and dispersed to separate its spectral information and then imaged on a detector. The main ERIS-SPIFFIER data reduction problems to solve are the following.

- Correct for the detector signature: bad pixels, detector contribution to the measured signal, flat fielding (correct pixel to pixel gain variations and relative slitlet throughput differences), correct geometric distortions.
- Perform the wavelength calibration.
- Reconstruct the image FOV from the 32 image slices in a format which contains both the spatial and the spectral information.
- Devise proper calibrations and observations to be able to properly correct the emission from the sky, from the instrument and from the telescope which are very strong in the NIR. This requires to take sky frames together with the object frames in the night observations, daily calibrations with the flat lamp switched on and off, and possibly dark frames.

In the following description we also indicate in parenthesis for each frame the corresponding PRO.CATG. To locate the detector bad pixels one uses a bad pixel map. A master bad pixel map resulting from the combination of a set of (different) bad pixel maps is generated by the master flat recipe. This master pixel map describes the main cosmetics of the ERIS IFU detector, and its effects in the reconstructed cube is shown in figure 11.2. Hot pixels will be determined on dark frames (BPM_DARK). Non-linear response pixels are instead indicated by a bad pixel map (BPM_DETLIN) obtained by evaluating the pixel response of a set of flat exposures of increasing intensity. Other bad pixels (BPM_DIST) are determined processing North-South data.

A master flat field (MASTER_FLAT) generated from a set of raw flat fields, is used to correct the different detector pixel sensitivities. It is known that the image sliced and projected on the detector is affected by distortions. The eris_ifu_distortion recipe computes the distortions (DISTORTION) and the slitlet positions (SLITLET_POS). It uses a set of raw frames where only the first column of each slitlet is illuminated through a slit. In addition flats and arcs are taken within the north south template as required to reduce the data. A set of “on” and “off” arc lamp frames, a reference line table, a master flat field, the optical distortions map and a good guess of the
slitlet positions are input of the wavecal recipe. This recipe determines a wavelength map and obtains a better computation of the left edge position of each slitlet.

Science (or PSF or telluric standard) frames are corrected for sky background, flat field, distortions, and calibrated in wavelength. In order to correct the effect of instrument flexures during night observations, the wavelength calibration is corrected using the OH sky lines.

The sorted slitlets are then stacked in a cube, taking into account the relative distances of the position of the edge of each slitlet to the one of the first slitlet, with a final image realignment to get sub pixel accuracy. The final product is thus a 3D data cube where the full spatial information is stored along the X and Y directions, and the wavelength information is stored along the Z direction. Each plane of the cube is a monochromatic reconstruction of the ERIS-SPIFFIER FOV.

The north south test template traces each of the 32 slitlets by only one fibre exposure; therefore non linearities of the image scale within the 64 pixel of a single slitlet are currently not corrected and could cause minor slice to slit ripples in the reconstructed cube.
11.3 Required input data

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

- **Reference files:**
  - A list of arc lamp emission lines containing vacuum wavelengths and predicted intensities for wavelength calibration.
  - The WAVE_SET UP table as input of eris_ifu_distortion and eris_ifu_wavecal to specify parameters peculiar of the wavelength calibration algorithm.

- **Raw frames:**
  - Linearity flat frames, to determine a map of non linear bad pixels.
  - Darks, to determine master darks.
  - Flat fields, to determine master flats.
  - Slit-Fibre frames, to trace the first column of each slitlet and, using also on/off lamp flats and arc lamp frames, to compute the optical distortions and slitlet distances.
  - Arc lamps, to perform the wavelength calibration.
  - Sky frames, to evaluate and subtract the strong and time-variable NIR sky emission.
  - Flux STD star frames, to compute the instrument response and efficiency.
– PSF standards, to evaluate the Strehl.
– Science frames, to finally do science.

• Calibration data products.
  – Bad pixel maps, to correct for the detector defects.
  – Distortion coefficients, to correct for the optical distortions.
  – Master flats, to correct for different detector pixel efficiencies.
  – Master darks, to correct for the instrument bias if no off or sky frames are available.
  – Slitlet positions, to be able to properly reconstruct a cube.
  – Wavelength maps, to obtain a cube calibrated in wavelength.

Calibration data products can be generated from raw data using the pipeline recipes. Alternatively the user may use calibration products obtained from the ESO archive or from the ESO Data Flow Operation department. Master bad pixel maps, the bad pixel maps coming from the standard flats, the master flats, the slitlets position table, the wavelength map depend from the observed band and instrument's pre-optic. Bad pixel maps coming from the non linearity test, distortion tables, slitlet distances, reference line tables depend only on the observed band. The reference bad pixel map, and master darks depend neither from the band nor the pre-optic. Science data requiring master dark need to have matching values of the FITS keyword DET SEQ1 DIT.

11.4 Reduction cascade

The ERIS-SPIFFIER data reduction follows the following sequence. A short description of the available recipes is given in section 7.1. In parenthesis we provide the value of the DO category corresponding to each frame.

• Run eris_ifu_detlin on a set of flats with increasing intensity (LINEARITY_LAMP) to determine the non linearity pixels bad pixel map (BPM_DETLIN).

• Run eris_ifu_dark on a set of raw darks (DARK) to determine the master dark (MASTER_DARK) and the hot pixels bad pixel map (BPM_DARK). This map depends on DET SEQ1 DIT.

• Run eris_ifu_flat on a set of standard flat fields (FLAT_LAMP), the BPM_DETLIN, BPM_DIST and the BPM_DARK, to determine the master bad pixel map (BPM_FLAT) and the master lamp flat (MASTER_FLAT).

• Run eris_ifu_distortion on a set of fibre flats (FIBRE_NS), on/off arc lamps (WAVE_NS) and on/off lamp flats (FLAT_NS), using a reference line table (REF_LINE_ARC) to determine the optical distortions (DISTORTION) and the slitlet distances (SLITLET_POS). To set a few data reduction parameters which depends from the observed band and used instrument pre-optics the user has also to provide in input a WAVE_SETUP table frame.

• Run eris_ifu_wavecal on a set of arc lamp frames (WAVE_LAMP), a MASTER_FLAT, recommended to remove long and short effects on the position of the arc lines before they are used to determine the wavelength solution, a DISTORTION, to determine the wavelength map (WAVE_MAP). To use a first guess solution, a list of reference arc lines, and to set a few data reduction parameters which depends from the
observed band and used instrument pre-optics the user has also to provide in input FIRST_WAVE_FIT, REF_LINE_ARC, a WAVE_SETUP.

• Run eris_ifu_stdstar on a reference flux standard (STD_FLUX) and a BPM_FLAT, a MASTER_FLAT, a DISTORTION, a SLITLET_POS, and a WAVE_MAP, and additional static calibrations, the EXTCOEFF_TABLE, the RESP_FIT_POINTS_CATALOG the TELL_MOD_CATALOG, the FIT AREAS, the QUALITY AREAS, the EFFICIENCY WINDOWS, the FLUX_STD_CATALOG, the HIGH_ABS_REGIONS to reduce the flux standards and get information on the instrument response and efficiency.

• Run eris_ifu_stdstar on PSF (or other STD) standards and a BPM_FLAT, a MASTER_FLAT, a DISTORTION, a SLITLET_POS, and a WAVE_MAP, a RESPONSE, and a EXTCOEFF_TABLE, to reduce the PSF (STD) standard and get information on the instrument's Strehl.

• Run eris_ifu_jitter on your scientific object (OBJ) and (if available) sky (SKY_OBJ) data and a BPM_FLAT, a MASTER_FLAT, a DISTORTION, a SLITLET_DISTANCES, a SLIT_POS and a WAVE_MAP the EXTCOEFF_TABLE, RESPONSE, to reduce science observations.

Examples of set of frames input for each recipe are provided in section 12.
12 Recipe Reference

12.1 Common Recipe Parameters

The following parameters can be applied to all recipes.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>product_depth</td>
<td>integer</td>
<td>0</td>
<td>set what kind of products are created (0 minimal, 1 more, ...)</td>
</tr>
<tr>
<td>line_corr</td>
<td>boolean</td>
<td>FALSE</td>
<td>If TRUE raw exposure image line correction will be applied.</td>
</tr>
<tr>
<td>col_corr</td>
<td>boolean</td>
<td>FALSE</td>
<td>If TRUE raw exposure image column correction will be applied.</td>
</tr>
<tr>
<td>crh_corr</td>
<td>boolean</td>
<td>FALSE</td>
<td>If TRUE raw exposure image cosmic ray hit correction will be applied.</td>
</tr>
<tr>
<td>crh_detection</td>
<td>boolean</td>
<td>FALSE</td>
<td>If TRUE raw exposure image cosmic ray hit detection will be applied.</td>
</tr>
<tr>
<td>crh.f_lim</td>
<td>double</td>
<td>2.0</td>
<td>Minimum contrast between the Laplacian image and the fine structure image that a point must have to be flagged as cosmics.</td>
</tr>
<tr>
<td>crh.max_iter</td>
<td>int</td>
<td>3</td>
<td>Maximum number of algorithm iterations.</td>
</tr>
</tbody>
</table>

**product_depth** Defines which output products will be generated.

- 0 = PD_SCI Only the main science or calibration products are created.
- 1 = PD_AUX Additional products will be create which either help to evaluate the final results or which are saving important intermediate steps.
- 2 = PD_ALL All products which might help the pipeline user are created.
- 3 = PD_DBG In addition products which might help the pipeline developer are created.

**line_corr** For SINFONI/SPIFFIER the read out electronics subtracts from all pixels in a certain row the mean of the 4 referenc pixels to the left and the right of the frame. This is done on purpose in order to get rid of a row-number dependant bias. However, whenever a hot pixel appears among the reference pixels, this creates too dark lines in the image. To correct for this the mean of the reference pixels is added to the whole row and then the median of the reference pixels is subtracted.

**col_corr** For SINFONI/SPIFFIER there can be artefacts in the vertical direction of the raw exposure images. One of the pattern is that every second column is too bright and every other too dark. The problem can be correct by subtracting the value of reference pixels which are on the detector for each column: Calculate the median of the first 4 and the last 4 pixels of each column and subtract the result from all pixels in this column.

**crh_corr** Enable cosmic ray hit correction (not yet implemented)

**crh_detection** Enable cosmic ray hit detection (not yet implemented)

**crh.sigma_lim** Refers to $\sigma_{lim}$ in the paper van Dokkum, PASP,113,2001,p1420-27

**crh.f_lim** Refers to $f_{lim}$ in the paper van Dokkum, PASP,113,2001,p1420-27

**crh.max_iter** Maximum number of iteration for CRH detection
12.2 eris_ifu_dark

12.2.1 Description

This recipe performs ERIS/SPIFFIER dark frames data reduction. It calculates the master dark frame and a bad pixel mask.

It is required to provide three or more dark exposures to produce a reasonable master dark with associated noise.

Two different bad pixel mask are generated selected by the \textit{–bpm.method} recipe parameter:

2D For the 2D bad pixel mask the master dark image is searched for outliers.

3D For the 3D bad pixel mask the provided dark exposures are stacked. Then outliers in the variance of each pixel are determined.

12.2.2 Input Frames

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARK</td>
<td>raw</td>
<td>&gt;= 3</td>
<td>Raw dark exposure</td>
</tr>
</tbody>
</table>

12.2.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Product Depth</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_dark_master_dark.fits</td>
<td>MASTER_DARK_IFU</td>
<td>0</td>
<td>Master dark frame (data, noise, DQI)</td>
</tr>
<tr>
<td>eris_ifu_dark_bpm.fits</td>
<td>BPM_DARK</td>
<td>0</td>
<td>Bad pixel mask</td>
</tr>
<tr>
<td>eris_ifu_dark_bpm2d.fits</td>
<td>DARK_BPM2D</td>
<td>1</td>
<td>2D bad pixel mask</td>
</tr>
<tr>
<td>eris_ifu_dark_bpm3d.fits</td>
<td>DARK_BPM3D</td>
<td>1</td>
<td>3D bad pixel mask</td>
</tr>
<tr>
<td>eris_ifu_dark_contribMap.fits</td>
<td>DARK_CONTRIBMAP</td>
<td>2</td>
<td>contribution map from image list collapse</td>
</tr>
<tr>
<td>eris_ifu_dark_dbg_image.fits</td>
<td></td>
<td>3</td>
<td>master dark image (single FITS image)</td>
</tr>
<tr>
<td>eris_ifu_dark_dbg_cube.fits</td>
<td></td>
<td>3</td>
<td>input dark images as stack (FITS image list)</td>
</tr>
<tr>
<td>eris_ifu_dark_dbg_3d.fits</td>
<td></td>
<td>3</td>
<td>Detailed output of 3D bad pixel detection</td>
</tr>
</tbody>
</table>

\textbf{eris_ifu_dark_master_dark.fits (MASTER_DARK_IFU)} This is the master dark frame in the format with three FITS extensions: data, noise and data quality. The data quality extension is an integer image with following bit encodings:
### bit number | bit value | description
--- | --- | ---
4 | 8 | saturated pixel
5 | 16 | general bad pixel (the first/last 4 rows/columns)
7 | 64 | bad pixel from BPM 2D
8 | 128 | bad pixel from BPM 3D

**eris_ifu_dark_bpm.fits** *(BPM_DARK)*  Bad pixel mask holding all types of bad pixels

**eris_ifu_dark_bpm2d.fits** *(DARK_BPM2D)*  2D bad pixel mask only

**eris_ifu_dark_bpm3d.fits** *(DARK_BPM3D)*  3D bad pixel mask only

**eris_ifu_dark_contribMap.fits** *(DARK_CONTRIBUTMAP)*  Contribution map from image list collapse

**eris_ifu_dark_dbg_image.fits**  Master dark image as a single FITS image

**eris_ifu_dark_dbg_cube.fits**  A cube (image stack) of the input raw dark exposures is the primary FITS data section, the first extension stores a cube of the bad pixel masks created by reading the images (edge and saturated pixels) and the second extension contains an image of the summed bad pixel masks.

**eris_ifu_dark_dbg_3d.fits**  The primary FITS data is an image where each pixel holds a count how many input frames are found too be bad for this pixel. The first extension hold a bit pattern image where the a set bit indicates the frame number where this pixel was detected as bad.

### 12.2.4 Recipe Parameters

Please see also the description for common recipe parameters in section 12.1 (page 50)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpm.method</td>
<td>string enum</td>
<td>2d3d</td>
<td>Specifies the VLT instrument 2d,3d,2d3d</td>
</tr>
<tr>
<td>collapse.method</td>
<td>string enum</td>
<td>MEDIAN</td>
<td>Method used for collapsing the data</td>
</tr>
<tr>
<td>collapse.sigclip.kappa-low</td>
<td>double</td>
<td>3</td>
<td>Low kappa factor for kappa-sigma clipping algorithm</td>
</tr>
<tr>
<td>collapse.sigclip.kappa-high</td>
<td>double</td>
<td>3</td>
<td>High kappa factor for kappa-sigma clipping algorithm</td>
</tr>
<tr>
<td>collapse.sigclip.niter</td>
<td>int</td>
<td>5</td>
<td>Maximum number of clipping iterations for kappa-sigma clipping</td>
</tr>
<tr>
<td>collapse.minmax.nlow</td>
<td>double</td>
<td>1</td>
<td>Low number of pixels to reject for the minmax clipping algorithm</td>
</tr>
<tr>
<td>collapse.minmax.nhigh</td>
<td>double</td>
<td>1</td>
<td>High number of pixels to reject for the minmax clipping algorithm</td>
</tr>
</tbody>
</table>

continued on next page
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collapse.mode.histo-min</td>
<td>double</td>
<td>10</td>
<td>Minimum pixel value to accept for mode computation</td>
</tr>
<tr>
<td>collapse.mode.histo-max</td>
<td>double</td>
<td>1</td>
<td>Maximum pixel value to accept for mode computation</td>
</tr>
<tr>
<td>collapse.mode.bin-size</td>
<td>double</td>
<td>0</td>
<td>Binsize of the histogram</td>
</tr>
<tr>
<td>collapse.mode.method</td>
<td>string enum</td>
<td>MEDIAN</td>
<td>Mode method (algorithm) to use</td>
</tr>
<tr>
<td>collapse.mode.error-niter</td>
<td>int</td>
<td>0</td>
<td>Iterations to compute the mode error</td>
</tr>
<tr>
<td>2dBadPix.method</td>
<td>string enum</td>
<td>FILTER</td>
<td>Method used</td>
</tr>
<tr>
<td>2dBadPix.legendre.kappa-low</td>
<td>double</td>
<td>10</td>
<td>Low RMS scaling factor for image thresholding</td>
</tr>
<tr>
<td>2dBadPix.legendre.kappa-high</td>
<td>double</td>
<td>10</td>
<td>High RMS scaling factor for image thresholding</td>
</tr>
<tr>
<td>2dBadPix.legendre.maxiter</td>
<td>int</td>
<td>3</td>
<td>Maximum number of algorithm iterations</td>
</tr>
<tr>
<td>2dBadPix.legendre.steps-x</td>
<td>int</td>
<td>20</td>
<td>Number of image sampling points in x-dir for fitting</td>
</tr>
<tr>
<td>2dBadPix.legendre.steps-y</td>
<td>int</td>
<td>20</td>
<td>Number of image sampling points in y-dir for fitting</td>
</tr>
<tr>
<td>2dBadPix.filter.kappa-low</td>
<td>int</td>
<td>10</td>
<td>X size of the median box around sampling points</td>
</tr>
<tr>
<td>2dBadPix.filter.kappa-high</td>
<td>int</td>
<td>10</td>
<td>Y size of the median box around sampling points</td>
</tr>
<tr>
<td>2dBadPix.filter.maxiter</td>
<td>int</td>
<td>2</td>
<td>Order of x polynomial for the fit</td>
</tr>
<tr>
<td>2dBadPix.filter.smooth-x</td>
<td>int</td>
<td>5</td>
<td>Kernel y size of the smoothing filter</td>
</tr>
<tr>
<td>2dBadPix.filter.smooth-y</td>
<td>int</td>
<td>5</td>
<td>Kernel y size of the image smoothing filter</td>
</tr>
<tr>
<td>3dBadPix.kappa-low</td>
<td>double</td>
<td>12</td>
<td>Low RMS scaling factor for image thresholding</td>
</tr>
<tr>
<td>3dBadPix.kappa-high</td>
<td>double</td>
<td>12</td>
<td>High RMS scaling factor for image thresholding</td>
</tr>
<tr>
<td>3dBadPix.method</td>
<td>string enum</td>
<td>MEDIAN</td>
<td>Thresholding method to use for bpm detection</td>
</tr>
<tr>
<td>qc_ron_xmin</td>
<td>int range</td>
<td>9</td>
<td>qc_ron_xmin</td>
</tr>
<tr>
<td>qc_ron_xmax</td>
<td>int range</td>
<td>2040</td>
<td>qc_ron_xmax</td>
</tr>
<tr>
<td>qc_ron_ymin</td>
<td>int range</td>
<td>9</td>
<td>qc_ron_ymin</td>
</tr>
<tr>
<td>qc_ron_ymax</td>
<td>int range</td>
<td>2040</td>
<td>qc_ron_ymax</td>
</tr>
</tbody>
</table>
12.2.5 Quality Control Parameters

QC.DARK.NBADPIX  
Total number of bad pixels but border pixels

QC.DARK.BPIXFRAC  
Fraction of bad pixels to total number of pixels

QC.DARK.NBADPIX2D  
Saturated pixels count

QC.DARK.NBADPIX3D  
Dark 2D bad pixels count

QC.DARK.BPIXFRAC2D  
Dark 3D bad pixels count

QC.DARK.BPIXFRAC3D  
Fraction of 2D bad pixel to total number of pixels

QC.MASTERDARK.MEAN  
[ADU] Clean mean of master dark image

QC.MASTERDARK.STDEV  
[ADU] Clean stdev of master dark image

QC.DARKMED.AVE  
[ADU] Average of raw darks medians

QC.DARKMED.STDEV  
[ADU] Read Out Noise

QC.DARKFPN  
Fixed Pattern Noise of combined frames

QC.RON  
[ADU] Read Out Noise

QC.RONRMS  
[ADU] Error on Read Out Noise

QC.RONi  
[ADU] Read Out Noise

QC.NRONS  
Number of RON frames

QC.DARKMED.AVE  
Calculate median of all raw dark exposures and return the average of the sequence

QC.DARKMED.STDEV  
Calculate median of all raw dark exposures and return the STDEV of the sequence

QC.DARKFPN  
Read out noise (see Readout Noise Computation, page 80) using the master dark frame

QC.RON  
Read out noise (see Readout Noise Computation, page 80) using the master dark frame

QC.RONRMS  
STDEV of the read out noise (see Readout Noise Computation, page 80) using the master dark frame

QC.RONi  
Read out noise (see Readout Noise Computation, page 80) using the difference of input frame \(i\) - input frame \(i+1\)
12.3 eris_ifu_detlin

12.3.1 Description

This recipe is used to determine a map of detector pixels that do not have a linear response to input light. The recipe input a set of (Lamp-on and Lamp-off) input frames obtained with increasing DET SEQ1 DIT.

12.3.2 Input Frames

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINEARITY_LAMP</td>
<td>raw</td>
<td>24 (min)</td>
<td>Raw linearity on and off frames</td>
</tr>
</tbody>
</table>

12.3.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_detlin_gain_info.fits</td>
<td>GAIN_INFO</td>
<td>Table with information on gain comutation</td>
</tr>
<tr>
<td>eris_ifu_detlin_bpm.fits</td>
<td>BPM_DETLIN</td>
<td>Bad pixel mask frame</td>
</tr>
<tr>
<td>eris_ifu_detlin_bpm_filt.fits</td>
<td>BPM_DETLIN_FILT</td>
<td>Filtered bad pixel mask frame</td>
</tr>
</tbody>
</table>

12.3.4 Quality Control Parameters

QC.LINEARITY.COEFFi

Linearity coefficient value

QC.LINEARITY.COEFFi.ERR

Linearity coefficient error value

QC.LINEARITY.NBADPIX

Number of bad pixels

QC.LINEARITY.BPIXFRAC

Fraction of bad pixels to total number of pixels

12.3.5 Recipe Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instrument</td>
<td>string</td>
<td>ERIS</td>
<td>Specifies the VLT instrument ERIS,SINFONI,NONE. &lt;ERIS</td>
</tr>
<tr>
<td>product_depth</td>
<td>int</td>
<td>0</td>
<td>Specifies the product output depth instrument (&gt;0 for auxillary products).</td>
</tr>
<tr>
<td>degree</td>
<td>int</td>
<td>2</td>
<td>Degree of polynomial to fit.</td>
</tr>
</tbody>
</table>

continued on next page
### 12.4 eris_ifu_distortion

#### 12.4.1 Description

This recipe allows to correct for raw detector image distortions in X and Y directions. For more details, please see Section 13.2.3.

#### 12.4.2 Input Frames

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pval</td>
<td>double</td>
<td>-1.0</td>
<td>p-value threshold (in percent). Fits with a p-value below this threshold are considered bad pixels.</td>
</tr>
<tr>
<td>rel-chi-low</td>
<td>double</td>
<td>8.0</td>
<td>Relative chi threshold. Pixels with a chi value smaller than mean - rel-threshold * stdev-of-chi are considered bad pixels.</td>
</tr>
<tr>
<td>rel-chi-high</td>
<td>double</td>
<td>8.0</td>
<td>Relative chi threshold. Pixels with a chi value larger than mean + rel-threshold * stdev-of-chi are considered bad pixels.</td>
</tr>
<tr>
<td>rel-coef-low</td>
<td>double</td>
<td>-1.0</td>
<td>Relative fit coefficient threshold. Pixels with a coefficient value smaller than mean +- rel-threshold * stdev-of-coeff are considered bad pixels.</td>
</tr>
<tr>
<td>rel-coef-high</td>
<td>double</td>
<td>-1.0</td>
<td>Relative fit coefficient threshold. Pixels with a coefficient value larger than mean +- rel-threshold * stdev-of-coeff are considered bad pixels.</td>
</tr>
<tr>
<td>pfx</td>
<td>int</td>
<td>3</td>
<td>X Size of the post filtering kernel.</td>
</tr>
<tr>
<td>ply</td>
<td>int</td>
<td>3</td>
<td>Y Size of the post filtering kernel.</td>
</tr>
<tr>
<td>pfm</td>
<td>string</td>
<td>closing</td>
<td>Post filtering mode. &lt;closing</td>
</tr>
<tr>
<td>stack-method</td>
<td>string</td>
<td>average, median, ksigma</td>
<td>Stacking method</td>
</tr>
<tr>
<td>clip-low</td>
<td>double</td>
<td>3</td>
<td>Low sigma threshold for ksigma pixel rejection method</td>
</tr>
<tr>
<td>clip-high</td>
<td>double</td>
<td>3</td>
<td>High sigma threshold for ksigma pixel rejection method</td>
</tr>
</tbody>
</table>
### 12.4.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_distortion_bpm.fits</td>
<td>BPM_DIST</td>
<td>Bad pixel mask frame</td>
</tr>
<tr>
<td>eris_ifu_distortion_slitlet_pos.fits</td>
<td>SLITLET_POS</td>
<td>Table frame with slitlet positions</td>
</tr>
<tr>
<td>eris_ifu_distortion_distortion.fits</td>
<td>DISTORTION</td>
<td>Table frame with distortions</td>
</tr>
</tbody>
</table>

### 12.4.4 Quality Control Parameters

- **QC.FLAT.SAT.NCOUNTS**: nr. saturated pixels of master flat
- **QC.SPECFLAT.OFFFLUX**: [ADU] average flux off frames
- **QC.SC.SPECFLAT.NCNTSAVG**: [ADU] average counts
- **QC.SC.SPECFLAT.NCNTSSSTD**: [ADU] stdev counts
- **QC.LFLAT.FPNi**: [ADU] Fixed Pattern Noise of combined frames
- **QC.FLAT.NBADPIX**: Number of bad pixels
- **QC.FLAT.BPIXFRAC**: Fraction of bad pixels to total
- **QC.FLAT.STDDEV**: [ADU] Std deviation on flat image
- **QC.FLAT.MEAN**: [ADU] Mean value on flat image
- **QC.FLAT.MEDIAN**: [ADU] Median value on flat image
- **QC.DISTORTION.NBADPIX**: Number of bad pixels
- **QC.DISTORTION.BPIXFRAC**: Fraction of bad pixels to total

### 12.4.5 Recipe Parameters

- clip-low & double & \textbf{3.} & 
  Low sigma threshold for \textit{ksigma} pixel rejection method $ackslash$
- **--instrument**: Specifies the VLT instrument {ERIS,SINFONI,NONE}. <ERIS | SINFONI | NONE> [NONE]
- **--product_depth**: Specifies the product output depth instrument (>0 for auxiliary products). [0]
- **--bpm.method**: Specifies the VLT instrument {2d,3d,2d3d}. <2d | 3d | 2d3d | none> [2d]
- **--collapse.method**: Method used for collapsing the data. <MEAN | WEIGHTED_MEAN | MEDIAN | SIGCLIP | MINMAX | --collapse.sigclip.kappa-low**: Low kappa factor for kappa-sigma clipping algorithm. [3.0]
--collapse.sigclip.kappa-high : High kappa factor for kappa-sigma clipping algorithm. [3.0]
--collapse.sigclip.niter : Maximum number of clipping iterations for kappa-sigma clipping. [5]
--collapse.minmax.nlow : Low number of pixels to reject for the minmax clipping algorithm. [1.0]
--collapse.minmax.nhigh : High number of pixels to reject for the minmax clipping algorithm. [1.0]
--collapse.mode.histo-min : Minimum pixel value to accept for mode computation. [10.0]
--collapse.mode.histo-max : Maximum pixel value to accept for mode computation. [1.0]
--collapse.mode.bin-size : Binsize of the histogram. [0.0]
--collapse.mode.method : Mode method (algorithm) to use. <MEDIAN | WEIGHTED | FIT> [MEDIAN]
--collapse.mode.error-niter : Iterations to compute the mode error. [0]

--2dBadPix.method : Method used. <FILTER | LEGENDRE> [LEGENDRE]
--2dBadPix.legendre.kappa-low : Low RMS scaling factor for image thresholding. [10.0]
--2dBadPix.legendre.kappa-high : High RMS scaling factor for image thresholding. [10.0]
--2dBadPix.legendre.maxiter : Maximum number of algorithm iterations. [3]
--2dBadPix.legendre.steps-x : Number of image sampling points in x-dir for fitting. [20]
--2dBadPix.legendre.steps-y : Number of image sampling points in y-dir for fitting. [20]
--2dBadPix.legendre.filter-size-x : X size of the median box around sampling points. [10]
--2dBadPix.legendre.filter-size-y : Y size of the median box around sampling points. [10]
--2dBadPix.legendre.order-x : Order of x polynomial for the fit. [2]
--2dBadPix.legendre.order-y : Order of y polynomial for the fit. [2]
--2dBadPix.filter.kappa-low : Low RMS scaling factor for image thresholding. [10.0]
--2dBadPix.filter.kappa-high : High RMS scaling factor for image thresholding. [10.0]
--2dBadPix.filter.maxiter : Maximum number of algorithm iterations. [3]
--2dBadPix.filter.border : Border mode to use for the image smoothing filter (only for MEDIAN filter). <FILTER | NOP> [NOP]
--2dBadPix.filter.smooth-x : Kernel y size of the smoothing filter. [5]
--2dBadPix.filter.smooth-y : Kernel y size of the smoothing filter. [5]

--3dBadPix.kappa-low : Low RMS scaling factor for image thresholding. [12.0]
--3dBadPix.kappa-high : High RMS scaling factor for image thresholding. [12.0]

--3dBadPix.method : Thresholding method to use for bpm detection. <absolute | relative | error> [relative]
--flat.mode : Mode of flat-calculation. <segment | hdrl | fast> [segment]
--flat_lo.filter-size-x : Smoothing filter size in x-direction. [5]
--flat_lo.filter-size-y : Smoothing filter size in y-direction. [5]
--flat_lo.method : Method to use for the master flatfield calculation. <low | high> [low]
--flat_hi.filter-size-x : Smoothing filter size in x-direction. [7]
--flat_hi.filter-size-y : Smoothing filter size in y-direction. [7]

--qc_fpn_xmin1 : Fixed Pattern Noise: qc_fpn_xmin1. [512]
--qc_fpn_ymin1 : Fixed Pattern Noise: qc_fpn_ymin1. [512]
--qc_fpn_xmax1 : Fixed Pattern Noise: qc_fpn_xmax1. [1536]
--qc_fpn_ymax1 : Fixed Pattern Noise: qc_fpn_ymax1. [1536]
--qc_fpn_xmin2 : Fixed Pattern Noise: qc_fpn_xmin2. [1350]
--qc_fpn_ymin2 : Fixed Pattern Noise: qc_fpn_ymin2. [1000]
--qc_fpn_xmax2 : Fixed Pattern Noise: qc_fpn_xmax2. [1390]
--qc_fpn_ymax2 : Fixed Pattern Noise: qc_fpn_ymax2. [1200]

--distortion.arcs.thresh_factor : median_value(image)+ kappa*sigma is the minimum intensity threshold of accepted pixels into
--distortion.arcs.min_arclen_factor : factor which sets minimum arc length (1.0.3) [1.19]
--distortion.arcs.window_size : Size of window for low pass filter used in an horizontal low pass filter to remove arcs
--distortion.smooth_rad : Size of smoothing factor (1-11) used to prevent for possible intensity drops from detector
--wave.div : Divide masterflat from subtracted wave frame. [FALSE]
--pixel_saturation : Pixel saturation level. [1.8e+04]
12.5 eris_ifu_flat

12.5.1 Description

This recipe allows to determine a master flat to correct for the pixel-to-pixel and larger spatial scale detector responsiveness, determine a map of dead pixels, and combine this map with the other input bad pixel maps.

12.5.2 Input Frames

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAT_LAMP</td>
<td>raw</td>
<td>10</td>
<td>Raw arc frame (5-on and 5-off)</td>
</tr>
<tr>
<td>BPM_DARK</td>
<td>cdb</td>
<td>1</td>
<td>hot pixel from dark frames</td>
</tr>
<tr>
<td>BPM_DETLIN</td>
<td>cdb</td>
<td>1</td>
<td>not linear pixels</td>
</tr>
<tr>
<td>BPM_DIST</td>
<td>cdb</td>
<td>1</td>
<td>bad pixels from distortion frames processing</td>
</tr>
</tbody>
</table>

12.5.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_flat_bpm.fits</td>
<td>BPM_FLAT</td>
<td>dead bad pixels map</td>
</tr>
<tr>
<td>eris_ifu_flat_master_flat.fits</td>
<td>MASTER_FLAT</td>
<td>Master flat</td>
</tr>
</tbody>
</table>

12.5.4 Quality Control Parameters

- QC.FLAT.SAT.NCOUNTS: nr. saturated pixels of master flat
- QC.SPECFLAT.OFFFLUX: [ADU] average flux off frames
- QC.SPECFLAT.NCMTSAVG: [ADU] average counts
- QC.SPECFLAT.NCMTSSD: [ADU] stdev counts
- QC.LFLAT.FPN1: [ADU] Fixed Pattern Noise
- QC.LFLAT.NBADPIX: Number of bad pixels
- QC.LFLAT.BPIXFRAC: Fraction of bad pixels to total pixel number
- QC.LFLAT.STDDEV: [ADU] Std deviation on flat image
- QC.LFLAT.MEAN: [ADU] Mean value on flat image
- QC.LFLAT.MEDIAN: [ADU] Median value on flat image

12.5.5 Recipe Parameters

- --instrument: Specifies the VLT instrument {ERIS, SINFONI, NONE}. [ERIS | SINFONI | NONE] [ERIS]
- --product_depth: Specifies the product output depth instrument (>0 for auxiliary products). [0]
- --bpm.method: Specifies the VLT instrument {2d, 3d, 2d3d}. [2d | 3d | 2d3d | none] [2d3d]
- --collapse.method: Method used for collapsing the data. [MEAN | WEIGHTED_MEAN | MEDIAN | SIGCLIP | MINMAX]
12.6 eris_ifu_wavecal

12.6.1 Description

This recipe allows to perform the wavelength calibration. For more details, please see Section 13.2.5.

12.6.2 Input Frames
### 12.6.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_wave_arclmg_stacked.fits</td>
<td>WAVE_LAMP_STACKED</td>
<td>Processed (stacked) arc lamp frame</td>
</tr>
<tr>
<td>eris_ifu_wave_arclmg_resampled.fits</td>
<td>WAVE_LAMP_STACKED_RESAMPLED</td>
<td>As previous but resampled</td>
</tr>
<tr>
<td>eris_ifu_wave_map.fits</td>
<td>WAVE_MAP</td>
<td>Wavelength map (pixel intensity equal to wavelength)</td>
</tr>
</tbody>
</table>

### 12.6.4 Quality Control Parameters

- QC.FRMONi.MEANFLUX [ADU] average of flux of image i
- QC.FRMONi.MEDIANFLUX [ADU] median of flux of image i
- QC.FRMONi.MAXFLUX [ADU] Max of flux of image i
- QC.FRMONi.NPIXSAT Number of saturated pixels of image i
- QC.FRMOFFi.MEANFLUX [ADU] Average of flux of image i
- QC.FRMOFFi.MEDIANFLUX [ADU] Median of flux of image i
- QC.FRMOFFi.MAXFLUX [ADU] Max of flux of image i
- QC.FRMOFFi.NPIXSAT Number of saturated pixels of image i
- QC.FRMDIFF.MEANFLUX [ADU] Average of flux of frame diff
- QC.FRMDIFF.MEDIANFLUX [ADU] Median of flux of frame diff
- QC.FRMDIFF.MAXFLUX [ADU] Max of flux of frame diff
- QC.FRMDIFF.NPIXSAT Number of saturated pixels of frame diff
- QC.SLITLETi.RESOL.AVG average resolution power for slitlet i
- QC.SLITLETi.RESOL.MED median resolution power for slitlet i
- QC.SLITLETi.RESOL.STD stdev resolution power for slitlet i
- QC.RESOL.AVG average resolution power
- QC.RESOL.MED median resolution power
- QC.RESOL.STD stdev resolution power
- QC.SLITLETi.FWHM.AVG [pix] average FWHM of found lines for slitlet i
- QC.SLITLETi.FWHM.MED [pix] median FWHM of found lines for slitlet i
- QC.SLITLETi.FWHM.STD [pix] stdev FWHM of found lines for slitlet i
- QC.FWHM.AVG [pix] average FWHM of found lines
- QC.FWHM.MED [pix] median FWHM of found lines
12.6.5 Recipe Parameters

```
--instrument : Specifies the VLT instrument {ERIS,SINFONI,NONE}. <ERIS | SINFONI | NONE> [ERIS]
--product_depth : Specifies the product output depth (>0 for auxiliary products). [0]
--line_corr : If TRUE raw exposure image line correction will be applied. [TRUE]
--col_corr : If TRUE raw exposure image column correction will be applied. [TRUE]
--crh_corr : If TRUE raw exposure image cosmic ray hit correction will be applied. [FALSE]
--crh_detection : If TRUE raw exposure image cosmic ray hit detection will be applied. [FALSE]
--crh.sigma_lim : Poisson fluctuation threshold to flag cosmics(see van Dokkum, PASP,113,2001,p1420-27) [5.0]
--crh.f_lim : Minimum contrast between the Laplacian image and the fine structure image that a point must have to be flagged as cosmics. [2.0]
--crh.max_iter : Maximum number of algorithm iterations. [3]
--pixel_saturation : Pixel saturation level. [1.8e+04]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instrument</td>
<td>string</td>
<td>ERIS</td>
<td>Specifies the VLT instrument {ERIS,SINFONI,NONE}. &lt;ERIS</td>
</tr>
<tr>
<td>product_depth</td>
<td>int</td>
<td>0</td>
<td>Specifies the product output depth (&gt;0 for auxiliary products). [0]</td>
</tr>
<tr>
<td>line_corr</td>
<td>boolean</td>
<td>TRUE</td>
<td>If TRUE raw exposure image line correction will be applied. [TRUE]</td>
</tr>
<tr>
<td>col_corr</td>
<td>boolean</td>
<td>TRUE</td>
<td>If TRUE raw exposure image column correction will be applied. [TRUE]</td>
</tr>
<tr>
<td>crh_corr</td>
<td>boolean</td>
<td>FALSE</td>
<td>If TRUE raw exposure image cosmic ray hit correction will be applied. [FALSE]</td>
</tr>
<tr>
<td>crh_detection</td>
<td>boolean</td>
<td>FALSE</td>
<td>If TRUE raw exposure image cosmic ray hit detection will be applied. [FALSE]</td>
</tr>
</tbody>
</table>

continued on next page
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crh.f_lim</td>
<td>double</td>
<td>2.0</td>
<td>Minimum contrast between the Laplacian image and the fine structure image that a point must have to be flagged as cosmics. [2.0]</td>
</tr>
<tr>
<td>crh.max_iter</td>
<td>int</td>
<td>3</td>
<td>Maximum number of algorithm iterations. [3]</td>
</tr>
</tbody>
</table>

### 12.7 eris_ifu_stdstar

#### 12.7.1 Description

#### 12.7.2 Input Frames

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD (*)</td>
<td>raw</td>
<td>1 (min)</td>
<td>Raw flux STD frame</td>
</tr>
<tr>
<td>SKY_STD (**)</td>
<td>raw</td>
<td>1 (min)</td>
<td>Raw sky frame</td>
</tr>
<tr>
<td>MASTER_FLAT</td>
<td>cdb</td>
<td>1</td>
<td>Master flat frame</td>
</tr>
<tr>
<td>BPM_FLAT</td>
<td>cdb</td>
<td>1</td>
<td>Master bad pixel map from flat frame</td>
</tr>
<tr>
<td>DISTORTION</td>
<td>cdb</td>
<td>1</td>
<td>Distortion frame</td>
</tr>
<tr>
<td>WAVE_MAP</td>
<td>cdb</td>
<td>1</td>
<td>Wavelength map frame</td>
</tr>
<tr>
<td>SLITLET_POS</td>
<td>cdb</td>
<td>1</td>
<td>Slitlet positions</td>
</tr>
<tr>
<td>OH_SPEC</td>
<td>static</td>
<td>1</td>
<td>Table with OH lines to get OH based wavelength solution</td>
</tr>
<tr>
<td>EXTCOEFF_TABLE</td>
<td>static</td>
<td>1</td>
<td>Atmospheric extinction table</td>
</tr>
<tr>
<td>RESP_FIT_POINTS_CATALOG</td>
<td>static</td>
<td>1</td>
<td>Catalog with response fit points</td>
</tr>
<tr>
<td>TELL_MOD_CATALOG</td>
<td>static</td>
<td>1</td>
<td>Catalogue with model spectra of atmospheric telluric absorption lines</td>
</tr>
<tr>
<td>FLUX_STD_CATALOG</td>
<td>static</td>
<td>1</td>
<td>Reference flux std catalog for efficiency and response computation</td>
</tr>
<tr>
<td>FIT AREAS</td>
<td>static</td>
<td>1</td>
<td>Table indicating region to fit response (band = J_low, H_low, K_low)</td>
</tr>
<tr>
<td>QUALITY AREAS</td>
<td>static</td>
<td>1</td>
<td>Table with regions to evaluate quality of fit response (band = J_low, H_low, K_low)</td>
</tr>
<tr>
<td>EFFICIENCY_WINDOWS</td>
<td>static</td>
<td>1</td>
<td>Table with regions used for efficiency computation (band = J_low, H_low, K_low)</td>
</tr>
</tbody>
</table>
### Frame Tag

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH_ABS_REGIONS</td>
<td>static</td>
<td>1</td>
<td>table indicating high absorption bands regions (band = J_low, H_low, K_low)</td>
</tr>
</tbody>
</table>

**Notes:**

(*) Alternative input tags are **STD_FLUX** or **PSF_CALIBRATOR** respectively for an input flux standard star or a PSF standard star.

(**) Alternative input tags are **SKY_STD_FLUX** or **SKY_PSF_CALIBRATOR** respectively for the sky associated to an input flux standard star or to a PSF standard star.

### 12.7.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_stdstar_dar_cube_001.fits</td>
<td>DAR_CORRECTED_CUBE</td>
<td>Object Data cube DAR corrected (created if -dar-corr=TRUE)</td>
</tr>
<tr>
<td>eris_ifu_stdstar_sky_cube_000.fits</td>
<td>SKY_CUBE</td>
<td>Sky Data cube</td>
</tr>
<tr>
<td>eris_ifu_stdstar_std_cube_001.fits</td>
<td>STD_CUBE</td>
<td>Object Data cube DAR</td>
</tr>
<tr>
<td>eris_ifu_stdstar_std_cube_coadd.fits (*)</td>
<td>STD_CUBE_COADD (*)</td>
<td>Combined data cube</td>
</tr>
<tr>
<td>eris_ifu_stdstar_std_cube_mean.fits (**')</td>
<td>STD_CUBE_MEAN ('')</td>
<td>Mean of combined cube</td>
</tr>
<tr>
<td>eris_ifu_stdstar_cube_median.fits</td>
<td>STD_CUBE_MEDIAN</td>
<td>Median of combined cube</td>
</tr>
<tr>
<td>eris_ifu_stdstar_extraction_mask.fits</td>
<td>EXTRACTION_MASK</td>
<td>Mask used to extract cube</td>
</tr>
<tr>
<td>eris_ifu_stdstar_spectrum.fits</td>
<td>SPECTRUM</td>
<td>extracted spectrum</td>
</tr>
<tr>
<td>eris_ifu_stdstar_response.fits</td>
<td>RESPONSE</td>
<td>computed response</td>
</tr>
<tr>
<td>eris_ifu_stdstar_spectrum_fluxcal.fits</td>
<td>SPECTRUM_FLUXCAL</td>
<td>Flux calibrated Std star spectrum</td>
</tr>
<tr>
<td>eris_ifu_stdstar_cube_fluxcal.fits</td>
<td>STD_CUBE_COADD_FLUXCAL</td>
<td>Flux calibrated Std star cube</td>
</tr>
<tr>
<td>eris_ifu_stdstar_no_flat_efficiency.fits</td>
<td>EFFICIENCY</td>
<td>computed efficiency</td>
</tr>
</tbody>
</table>

(*) In case of **STD_FLUX** input the product will be named `eris_ifu_stdstar_std_flux_cube_coadd.fits` and have PRO.CATG **STD_FLUX_CUBE_COADD**. In case of **PSF_CALIBRATOR** input the product will be named `eris_ifu_psf_flux_cube_coadd.fits` and have PRO.CATG **PSF_CUBE_COADD**.

(**) In case of **STD_FLUX** input the product will be named `eris_ifu_stdstar_std_flux_cube_mean.fits` and have PRO.CATG **STD_FLUX_CUBE_MEAN**. In case of **PSF_CALIBRATOR** input the product will be named `eris_ifu_psf_flux_cube_mean.fits` and have PRO.CATG **PSF_CUBE_MEAN**.

### 12.7.4 Quality Control Parameters

- QC.OH.LAMBDA.SHIFT.UM [um] OH based center lambda shift to (each) lamp calib frame
- QC.OH.LAMBDA.SHIFT.PIX [pix] OH based center lambda shift to (each) lamp calib frame
- QC.FLAT.SAT.NCOUNTS [] nr. saturated pixels of master flat
- QC.SPECFLAT.OFFFLUX [] average flux off frames
QC.SPECFLAT.NCNTSAVG [] average counts
QC.SPECFLAT.NCNTSSTD [] stdev counts
QC.LFLAT.FPN1 [] Fixed Pattern Noise of combin
QC.LFLAT.FPN2 [] Fixed Pattern Noise of combin
QC.FLAT.NBADPIX Number of bad pixels
QC.FLAT.BPIXFRAC Fraction of bad pixels to to
QC.RESP.WINi.WLMIN [um] Min window wavelength for resp co
QC.RESP.WINi.WLMAX [um] Max window wavelength for resp co
QC.RESP.WINi.MEAN Mean response on window
QC.RESP.WINi.MEDIAN Median response on window
QC.RESP.WINi.MIN Min response on window
QC.RESP.WINi.MAX Max response on window
QC.RESP.WINi.RMS RMS response on window
QC.RESP.WLMIN [um] Min wavelength for resp comp
QC.RESP.WLMAX [um] Max wavelength for resp comp
QC.RESP.MEAN Mean response
QC.RESP.MEDIAN Median response
QC.RESP.MIN Min response
QC.RESP.MAX Max response
QC.RESP.RMS RMS response
QC.FWHM.LLX [pix] Lower lext X pos for FWHM detection
QC.FWHM.LLY [pix] Lower lext Y pos for FWHM detection
QC.FWHM.HBX [pix] Half Box size for FWHM detection
QC.FWHM.CENTERX [pix] STD star X centroid position
QC.FWHM.CENTERY [pix] STD star Y centroid position
QC.FWHM.MAJ [pix] STD star FWHM on major axis size
QC.FWHM.MIN [pix] STD star FWHM on minor axis size
QC.CHECK1 Check on evaluation box
QC.FWHMX [pix] STD star FWHM on X direction
QC.FWHMY [pix] STD star FWHM on Y direction
QC.STREHL Computed Strehl
QC.STREHL.ERR Computed Strehl error
QC.EFF.WINi.WLMIN [um] Min window wavelength for eff comp
QC.EFF.WINi.WLMAX [um] Max window wavelength for eff comp
QC.EFF.WINi.MEAN Mean efficiency on window
QC.EFF.WINi.MEDIAN Median efficiency on window
QC.EFF.WINi.MIN Min efficiency on window
QC.EFF.WINi.MAX Max efficiency on window
QC.EFF.WINi.RMS RMS efficiency on window
QC.EFF.WINi.WLMIN [um] Min wavelength for eff comp
QC.EFF.WINi.WLMAX [um] Max wavelength for eff comp
QC.EFF.WINi.MEAN Mean efficiency
QC.EFF.WINi.MEDIAN Median efficiency
QC.EFF.WINi.MIN Min efficiency
QC.EFF.WINi.MAX Max efficiency
12.7.5 Recipe Parameters

- **--instrument**: Specifies the VLT instrument {ERIS, SINFONI, NONE}. (ERIS | SINFONI | NONE) [ERIS]
- **--product_depth**: Specifies the product output depth (>0 for auxiliary products). [0]
- **--line_corr**: If TRUE raw exposure image line correction will be applied. [TRUE]
- **--col_corr**: If TRUE raw exposure image column correction will be applied. [TRUE]
- **--crh_corr**: If TRUE raw exposure image cosmic ray hit correction will be applied. [FALSE]
- **--crh_detection**: If TRUE raw exposure image cosmic ray hit detection will be applied. [FALSE]
- **--crh.f_lim**: Minimum contrast between the Laplacian image and the fine structure image that a point must have to be flagged as cosmics. [2.0]
- **--crh_max_iter**: Maximum number of algorithm iterations. [3]
- **--pixel_saturation**: Pixel saturation level. [1.8e+04]
- **--sky_tweak**: Use modified sky cube for sky subtraction. [FALSE]
- **--discard_subband**: Ignore last sub-band in the sky tweaking. [FALSE]
- **--stretch**: Stretch sky in the sky tweaking. [FALSE]
- **--stretch_degree**: Stretch polynomial degree. [8]
- **--stretch_resampling**: Stretch resampling method (linear/spline). [spline]
- **--tbsub**: Subtract thermal background from input cube. [TRUE] (apply) or FALSE (don't apply). [TRUE]
- **--velocity_offset**: Specify velocity offset correction in km/s for lambda scale. [0.0]
- **--bpm_threshold**: Specify a threshold for the interpolated BPM values. [0.5]
- **--cube.slitlet-detection**: Specifies the slitlet detection: "DIST" slitlet distances as detected by distortion recipe or "EDGE" slitlet edges as detected be wavecalrecipe. (DIST | EDGE | GRID) [DIST]
- **--cube.first-col**: Specifies the first column offset in case the cube.slitlet-detection parameter is set to "DIST". [1.0]
- **--cube.fine-tune**: Specifies the row interpolation mode: 0 -> no interpolation, otherwise see eris_ifu_1d_interpolation. [-1]
- **--method**: Resampling method. (NEAREST | LINEAR | QUADRATIC | RENKA | DRIZZLE | LANCZOS) [LANCZOS]
- **--method.loop-distance**: Loop distance used by all (but NEAREST) methods to control the number of surrounding voxels that are taken into account. [1]
- **--method.use-errorweights**: Use additional weights of 1/err^2. [FALSE]
- **--method.renka.critical-radius**: Critical radius of the Renka method. [1.25]
- **--method.drizzle.downscale-y**: Drizzle down-scaling factor in y direction. [0.8]
- **--subtract-background**: Subtract median of the images in 2D only. [FALSE]
- **--extract-source**: If True try to extract a point source. [TRUE]
- **--mask_method**: Method to specify extraction mask: mask, position, max or fit. [max]
- **--center**: The centre of the circular mask (pixel). [32,32]
- **--radius**: The radius of the circular mask (pixel). [9.0]
- **--flux_calibrate**: If True flux calibrate the extracted spectrum and data cube. [TRUE]
- **--dar-corr**: Correct Differential Atmospheric Refraction (DAR). [TRUE]
- **--dar-shift-method**: The method to shift images for DAR correction. [0]
- **--dar-shift-length**: Kernel length for DAR shift image interpolation. [0.0]
- **--fwhm-factor**: Factor to find 2D-Gauss FWHM. The extraction box is: halfbox_x=halfbox_y=fwhm_factor*(fwhm_x+fwhm_y)*0.5. [5.0]
- **--max-cubes-centres-dist**: Maximum distance between cube centers to build a mosaic. Mosaic creation requires a lot of RAM. [240]
- **--cube.combine**: With multi-pointing observations combine cubes into a mosaic. [TRUE]
- **--derot_corr**: The first column offset to be applied on the science data for distortion correction. Typical value is within +/- 2 pixels. A default correction based on altitude and rotation angle will be applied. [auto]
- **--bpc_iter**: No. of iterations for bad pixel correction. [1]
- **--chop-nan**: If true chop cube planes with more than 50% NAN pixels. [FALSE]
- **--crea-phase3**: If true create phase3 compliant data products. [FALSE]
- **--compute-efficiency**: If True try to compute the efficiency from a STD star spectrum. [TRUE]
- **--compute-response**: If True try to compute the response from a STD star spectrum. [TRUE]
12.8 eris_ifu_jitter

12.8.1 Description

The eris_ifu_jitter recipes reduces scientific targets data by implementing following steps:

- Collect bad pixel masks
- Apply flat field correction (when a master flat image is provided)
- Resample the exposure frames to wavelength calibrated data cubes
- Correct for sky features
- Correct for differential atmospheric refraction (DAR)
- Resample multiple exposure to a single co-added cube
- Extract a source spectrum

Bad pixel masks

The recipe accepts several bad pixel masks (BPM_DARK, BPM_FLAT and BPM_DET LIN) and the data quality extension of the MASTER_FLAT image. All the bad pixel masked are or’ed so a bad pixel in any mask flags this pixel a bad in the final mask.

Flat fielding

When a MASTER_FLAT image is provided all exposure images will be divided by it.

Sky correction

The recipe checks the FITS header information of the exposure raw frames to find the optimum object-sky pairs. If several skies are available, the one closest in time to the object will be picked. If there is no sky available, there won’t be any sky subtraction applied. The order of the files in the SOF is irrelevant.

When the –sky_tweak parameter is set to 0 the sky exposure image is simply subtracted from the object exposure. Setting the –sky_tweak parameter to 1 activates the sky subtraction using the method of Ric Davies[3]. Here the sky cube is tweaked according to pixels in the object cube which represent the background signal. Then this adapted sky cube is subtracted from the object cube.

Differential atmospheric refraction (DAR) correction

Setting the dar-corr is set to TRUE the DAR correction will be enabled. With the –dar-shift-method parameter several methods can be chosen to shift the image planes of the cube.

The input cube type depends on the preprocessing:

- When DAR correction is enabled the DAR corrected cubes will be used.
- When sky tweaking is enabled the sky tweaked cubes will be used.

In case a dedicated sky frame is missing, the sky can still be modeled and corrected with the methods controlled by positive values of the parameter aj-method. We remind the reader to a description of that parameter for more details.
• Otherwise the object cubes will be used.

When there is just one object cube the co-added cube is just a copy of it. The resampling method and its parameters can be tuned using the method... recipes parameters.

**Spectrum extraction** TBW

### 12.8.2 Input Frames

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ</td>
<td>raw</td>
<td>1 ... n</td>
<td>Object exposure frame</td>
</tr>
<tr>
<td>SKY or SKY_OBJ</td>
<td>raw</td>
<td>1 ... n</td>
<td>Sky exposure frame</td>
</tr>
<tr>
<td>DISTORTION</td>
<td>cdb</td>
<td>1</td>
<td>Table with distortion correction parameters</td>
</tr>
<tr>
<td>WAVE_MAP</td>
<td>cdb</td>
<td>1</td>
<td>Wavelength calibration map</td>
</tr>
<tr>
<td>MASTER_DARK</td>
<td>cdb</td>
<td>[0,1]</td>
<td>Optional master dark image</td>
</tr>
<tr>
<td>MASTER_FLAT</td>
<td>cdb</td>
<td>[0,1]</td>
<td>Optional master flat image</td>
</tr>
<tr>
<td>BPM_DARK</td>
<td>cdb</td>
<td>[0,1]</td>
<td>Optional bad pixel mask</td>
</tr>
<tr>
<td>BPM_FLAT</td>
<td>cdb</td>
<td>[0,1]</td>
<td>Optional bad pixel mask</td>
</tr>
<tr>
<td>BPM_DETIN</td>
<td>cdb</td>
<td>[0,1]</td>
<td>Optional bad pixel mask</td>
</tr>
<tr>
<td>OH_SPEC</td>
<td>static</td>
<td>1</td>
<td>Vector holding OH lines spectrum</td>
</tr>
<tr>
<td>RESPONSE</td>
<td>cdb</td>
<td>[0,1]</td>
<td>Instrument response</td>
</tr>
<tr>
<td>EXTCOEFF_TABLE</td>
<td>static</td>
<td>[0,1]</td>
<td>Atmospheric extinction table</td>
</tr>
</tbody>
</table>

### 12.8.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Product Depth</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_jitter_obj_cube_nnn.fits</td>
<td>OBJECT_CUBE</td>
<td>0</td>
<td>Object data cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_sky_cube_nnn.fits</td>
<td>SKY_CUBE</td>
<td>0</td>
<td>Sky data cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_twk_cube_nnn.fits</td>
<td>SKY_TWEAKED_CUBE</td>
<td>0</td>
<td>Object data sky tweaked cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_dar_cube_nnn.fits</td>
<td>DAR_CORRECTED_CUBE</td>
<td>0</td>
<td>Object data cube DAR corrected</td>
</tr>
<tr>
<td>eris_ifu_jitter_dar_cube_coadd.fits</td>
<td>DAR_CORRECTED_CUBE_COADD</td>
<td>0</td>
<td>Coadded object cube, DAR corrected</td>
</tr>
<tr>
<td>eris_ifu_jitter_dar_cube_mean.fits</td>
<td>DAR_CORRECTED_CUBE_MEAN</td>
<td>0</td>
<td>Collapsed object cube, DAR corrected</td>
</tr>
<tr>
<td>eris_ifu_jitter_twk_cube_coadd.fits</td>
<td>SKY_TWEAKED_CUBE_COADD</td>
<td>0</td>
<td>Co-added object cube, &quot;sky tweaked&quot;</td>
</tr>
<tr>
<td>eris_ifu_jitter_twk_cube_mean.fits</td>
<td>SKY_TWEAKED_CUBE_MEAN</td>
<td>0</td>
<td>Collapsed object cube, &quot;sky tweaked&quot;</td>
</tr>
<tr>
<td>eris_ifu_jitter_obj_cube_coadd.fits</td>
<td>OBJECT_CUBE_COADD</td>
<td>0</td>
<td>Co-added object cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_obj_cube_mean.fits</td>
<td>OBJECT_CUBE_MEAN</td>
<td>0</td>
<td>Collapsed object cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_bpm_cube_nnn.fits</td>
<td>BPM_CUBE</td>
<td>0</td>
<td>Interpolated BPM of object/sky cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_cube_median.fits</td>
<td>STD_CUBE_MEDIAN</td>
<td>0</td>
<td>Median of co-added cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_extraction_mask.fits</td>
<td>EXTRACTION_MASK</td>
<td>0</td>
<td>Mask used to extract cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_spectrum.fits</td>
<td>SPECTRUM</td>
<td>0</td>
<td>Extracted spectrum</td>
</tr>
<tr>
<td>eris_ifu_jitter_spectrum_fluxcal.fits</td>
<td>SPECTRUM_FLUXCAL</td>
<td>0</td>
<td>Flux calibrated object spectrum</td>
</tr>
</tbody>
</table>

continued on next page


continued from previous page

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Product Depth</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_jitter_cube_fluxcal.fits</td>
<td>OBJECT_CUBE_COADD_FLUXCAL</td>
<td>0</td>
<td>Flux calibrated object cube</td>
</tr>
<tr>
<td>eris_ifu_jitter_cube_fluxcal_mean.fits</td>
<td>OBJECT_CUBE_COADD_FLUXCAL_MEAN</td>
<td>0</td>
<td>Flux calibrated object cube mean</td>
</tr>
<tr>
<td>eris_ifu_jitter_obj_exposure_map.fits</td>
<td>EXPOSURE_MAP</td>
<td>0</td>
<td>Exposure map</td>
</tr>
</tbody>
</table>

**eris_ifu_jitter_obj_cube_ nnn.fits (OBJECT_CUBE)** Reconstructed cube from the $n^{th}$ raw exposure frame specified in the SOF file (tagged as object).

**eris_ifu_jitter_sky_cube_ nnn.fits (SKY_CUBE)** Reconstructed cube from the $n^{th}$ raw exposure frame specified in the SOF file (tagged as sky).

**eris_ifu_jitter_twk_cube_ nnn.fits (SKY_TWEAKED_CUBE)** Reconstructed cube from the $n^{th}$ raw exposure frame specified in the SOF file with the sky tweak algorithm applied. This product is only generated when the $–sky\_tweak$ parameter is set to 1.

**eris_ifu_jitter_dar_cube_ nnn.fits (DAR_CORRECTED_CUBE)** Reconstructed cube from the $n^{th}$ raw exposure frame specified in the SOF file with the DAR correction applied. If the $–sky\_tweak$ parameter is set to one the sky tweak algorithm is also applied. This product is only generated when the $–dar\_corr$ parameter is set to TRUE.

**eris_ifu_jitter_dar_cube_coadd.fits (DAR_CORRECTED_CUBE_COADD)** Resampled cube using all DAR_CORRECTED_CUBE cubes as input. This product is only generated when both the $–dar\_corr$ is set to TRUE.

**eris_ifu_jitter_dar_cube_mean.fits (DAR_CORRECTED_CUBE_MEAN)** Image of the collapsed DAR_CORRECTED_CUBE_COADD cube calculating the average of each spaxel.

**eris_ifu_jitter_twk_cube_coadd.fits (SKY_TWEAKED_CUBE_COADD)** Resampled cube using all SKY_TWEAKED_CUBE cubes as input. This product is only generated when the $–sky\_tweak$ parameter is set to 1 and the $–dar\_corr$ parameter is set to FALSE.

**eris_ifu_jitter_twk_cube_mean.fits (SKY_TWEAKED_CUBE_MEAN)** Image of the collapsed SKY_TWEAKED_CUBE_COADD cube calculating the average of each spaxel.

**eris_ifu_jitter_obj_cube_coadd.fits (OBJECT_CUBE_COADD)** Resampled cube using all OBJECT_CUBE cubes as input.

**eris_ifu_jitter_obj_cube_mean.fits (OBJECT_CUBE_MEAN)** Image of the collapsed OBJECT_CUBE_COADD cube calculating the average of each spaxel.

**eris_ifu_jitter_bpm_cube_ nnn.fits (BPM_CUBE)**

**eris_ifu_jitter_cube_median.fits (STD_CUBE_MEDIAN)** Image of the collapsed OBJECT_CUBE_COADD cube calculating the average of each spaxel. This file is only created when the source spectrum extraction is requested as this image is the one to locate the source by a 2D Gauss fit.
eris_ifu_jitter_extraction_mask.fits (EXTRACTION_MASK)  Mask image indicating pixels used to extract the object.

eris_ifu_jitter_spectrum.fits (SPECTRUM)  The extracted spectrum.

eris_ifu_jitter_spectrum_fluxcal.fits (SPECTRUM_FLUXCAL)  The flux calibrated extracted spectrum.

eris_ifu_jitter_cube_fluxcal.fits (STD_CUBE_COADD_FLUXCAL)  The flux calibrated combined cube.

eris_ifu_jitter_cube_fluxcal_mean.fits (STD_CUBE_COADD_FLUXCAL_MEAN)  The flux calibrated combined cube mean.

eris_ifu_jitter_obj_cube_exposure_map.fits (EXPOSURE_MAP)  The a map indicating the median exposure time used for each cube plane exposed data point.

12.8.4 Recipe Parameters

We list here only the most important parameters. Please see also the description for common recipe parameters in section 12.1 (page 50)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sky_tweak</td>
<td>int</td>
<td>0</td>
<td>Use modified sky cube for sky subtraction.0: don’t apply, 1: Davies’ method, 2: Austrian method</td>
</tr>
<tr>
<td>skip_sky_oh_align</td>
<td>bool</td>
<td>False</td>
<td>Skip the OH alignment for the SKY (deprecated)4</td>
</tr>
<tr>
<td>discard_subband</td>
<td>bool</td>
<td>False</td>
<td>Ignore last sub-band in the sky tweaking</td>
</tr>
<tr>
<td>stretch</td>
<td>bool</td>
<td>False</td>
<td>Stretch sky in the sky tweaking</td>
</tr>
<tr>
<td>stretch_degree</td>
<td>int</td>
<td>8</td>
<td>Stretch polynomial degree</td>
</tr>
<tr>
<td>stretch_resampling</td>
<td>string</td>
<td>spline</td>
<td>Stretch resampling method (linear/spline)</td>
</tr>
<tr>
<td>tbsub</td>
<td>bool</td>
<td>True</td>
<td>Subtract thermal background from input cube.(TRUE (apply) or FALSE (don’t apply)</td>
</tr>
<tr>
<td>velocity_offset</td>
<td>double</td>
<td>0</td>
<td>Specify velocity offset correction in km/s for lambda scale</td>
</tr>
<tr>
<td>bpm_threshold</td>
<td>double</td>
<td>0.5</td>
<td>Specify a threshold for the interpolated BPM values</td>
</tr>
<tr>
<td>cube.slitlet-detection</td>
<td>string enum</td>
<td>DIST</td>
<td>Specifies the slitlet detection: “DIST” slitlet distances as detected by distortion recipe or “EDGE” slitlet edges as detected be wavecal-recipe</td>
</tr>
<tr>
<td>cube.first-col</td>
<td>double</td>
<td>1</td>
<td>Specifies the first column offset in case the cube.slitlet-detection parameter is set to “DIST”</td>
</tr>
<tr>
<td>cube.fine-tune</td>
<td>int</td>
<td>-1</td>
<td>Specifies the row interpolation mode: 0 -&gt; no interpolation, otherwise see eris_ifu_1d_interpolation</td>
</tr>
<tr>
<td>cube.combine</td>
<td>bool</td>
<td>True</td>
<td>With multi-pointing observations combine cubes into a mosaic.</td>
</tr>
</tbody>
</table>

continued on next page

4In case there are no OH lines detectable, the OH alignment can be skipped. (i.e. very bright object, K_LONG, small plate-scale, short DIT). As a result there will be an overall wavelength shift in the output cube which has to be corrected oneself.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>string enum</td>
<td>DRIZZLE</td>
<td>Resampling method</td>
</tr>
<tr>
<td>method.loop-distance</td>
<td>int</td>
<td>3</td>
<td>Loop distance used by all (but NEAREST) methods to control the number of surrounding voxels that are taken into account. A small value allows faster re-sampling but may not give good quality.</td>
</tr>
<tr>
<td>method.use-errorweights</td>
<td>bool</td>
<td>False</td>
<td>Use additional weights of $1/\text{err}^2$</td>
</tr>
<tr>
<td>method.renka.critical-radius</td>
<td>double</td>
<td>1.25</td>
<td>Critical radius of the Renka method</td>
</tr>
<tr>
<td>method.lanczos.kernel-size</td>
<td>int</td>
<td>2</td>
<td>Kernel size of the Lanczos method</td>
</tr>
<tr>
<td>method.drizzle.downscale-x</td>
<td>double</td>
<td>0.8</td>
<td>Drizzle down-scaling factor in x direction</td>
</tr>
<tr>
<td>method.drizzle.downscale-y</td>
<td>double</td>
<td>0.8</td>
<td>Drizzle down-scaling factor in y direction</td>
</tr>
<tr>
<td>method.drizzle.downscale-z</td>
<td>double</td>
<td>0.8</td>
<td>Drizzle down-scaling factor in wavelength direction</td>
</tr>
<tr>
<td>subtract-background</td>
<td>bool</td>
<td>False</td>
<td>Subtract median of the images in 2D only</td>
</tr>
<tr>
<td>fieldmargin</td>
<td>double</td>
<td>5</td>
<td>Add this margin/border (in percent) to the resampled image/cube</td>
</tr>
<tr>
<td>edge-trim</td>
<td>int</td>
<td>0</td>
<td>Number or pixels to trim for each plane of the input frames. It should be smaller than half image size</td>
</tr>
<tr>
<td>extract-source</td>
<td>bool</td>
<td>False</td>
<td>If True try to extract a point source</td>
</tr>
<tr>
<td>mask_method</td>
<td>string</td>
<td>max</td>
<td>Method to specify extraction mask : mask, position, max or fit</td>
</tr>
<tr>
<td>center</td>
<td>string</td>
<td>32,32</td>
<td>The centre of the circular mask (pixel)</td>
</tr>
<tr>
<td>radius</td>
<td>double</td>
<td>4</td>
<td>The radius of the circular mask (pixel)</td>
</tr>
<tr>
<td>flux-calibrate</td>
<td>bool</td>
<td>False</td>
<td>If True flux calibrate the extracted spectrum and data cube</td>
</tr>
<tr>
<td>dar-corr</td>
<td>bool</td>
<td>False</td>
<td>Correct Differential Atmospheric Refraction (DAR)</td>
</tr>
<tr>
<td>dar-shift-method</td>
<td>int range</td>
<td>0</td>
<td>The method to shift images for DAR correction</td>
</tr>
<tr>
<td>dar-shift-length</td>
<td>int</td>
<td>0</td>
<td>Kernel length for DAR shift image interpolation</td>
</tr>
<tr>
<td>dar-shift-radius</td>
<td>double</td>
<td>0</td>
<td>Kernel radius for DAR shift image interpolation</td>
</tr>
<tr>
<td>fwhm-factor</td>
<td>double</td>
<td>5</td>
<td>Factor to find 2D-Gauss FWHM. The extraction box is: half-box_x=halfbox_x/fwhm_factor*(fwhm_x+fwhm_y)*0.5.</td>
</tr>
<tr>
<td>max-cubes-centres-dist</td>
<td>int</td>
<td>240</td>
<td>Maximum distance between cube centers to build a mosaic. Mosaic creation requires a lot of RAM. Users may trim this value to fit RAM resources.</td>
</tr>
<tr>
<td>cubecombine</td>
<td>bool</td>
<td>TRUE</td>
<td>With multi-pointing observations combine cubes into a mosaic.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>derot_corr</td>
<td>string</td>
<td>auto</td>
<td>The first column offset to be applied on the science data for distortion correction. This effect is visible when the spectrum on one side of the cube wraps to the other side. Typical value is within +/- 2 pixels. A default correction based on altitude and rotation angle will be applied if this parameter set to auto; otherwise the user-specified number of the shift would be used.</td>
</tr>
<tr>
<td>aj-method</td>
<td>int</td>
<td>7</td>
<td>Method used to estimate the sky in case of missing sky frames. 0: no sky subtraction. 1: sky is taken from the next in MJD-OBS object frame. 2: sky is taken from the median of all input sky frames. 3: sky is taken from the mean of all input sky frames. 4: sky is taken from the median of user-specified rect. box; 5: sky is taken from the median of 4 regions at FOV edges; 6: sky is taken from the median of 8 regions at FOV edges; 7: sky is taken from each cube slice based on ks-clip based mask computed on collapsed cube; 8: sky is taken from each cube slice based on 25% clip based mask computed on collapsed cube.</td>
</tr>
<tr>
<td>sky-box-center</td>
<td>int,int</td>
<td>32,32</td>
<td>The centre of the rectangular sky region (pixel). Two integer numbers each in the range [1,64].</td>
</tr>
<tr>
<td>sky-box-width</td>
<td>int,int</td>
<td>1,1</td>
<td>The rectangular (small) sky region x,y half widths [pixel]. Two numbers each in the range [0,10].</td>
</tr>
<tr>
<td>sky-box-edges-margin</td>
<td>int, int</td>
<td>1,1</td>
<td>The (small) number of x, y pixels taken as margin to the FOV edges so that sky regions (aj-method=5,6) do not include pixels at the edges of the re-sampled image [pixel]. Two numbers each in the range [0,5].</td>
</tr>
<tr>
<td>sky-est-method</td>
<td>int</td>
<td>0</td>
<td>If the user has no input sky frame and the sky need to be estimated from user defined pixels set by aj-method this parameter allow to use the median (0) or the mean (1) [0].</td>
</tr>
<tr>
<td>sky-est-kappa</td>
<td>double</td>
<td>3.0</td>
<td>If the user has no input sky frame and the sky need to be estimated from user defined pixels set by aj-method this parameter allow to use the kappa of the kappa sigma clip iteration.</td>
</tr>
<tr>
<td>sky-est-niter</td>
<td>int</td>
<td>5</td>
<td>If the user has no input sky frame and the sky need to be estimated from user defined pixels set by aj-method this parameter allow to set the number of the kappa sigma clip iterations.</td>
</tr>
<tr>
<td>bpc_iter</td>
<td>int</td>
<td>1</td>
<td>No. of iterations for bad pixel correction.</td>
</tr>
<tr>
<td>chop-nan</td>
<td>boolean</td>
<td>FALSE</td>
<td>If true chop cube planes with more than 50% NAN pixels.</td>
</tr>
<tr>
<td>crea-phase3</td>
<td>boolean</td>
<td>FALSE</td>
<td>If true crea phase3 compliant data products.</td>
</tr>
</tbody>
</table>

dar-corr is set to TRUE the object cubes will be corrected for differential atmospheric refraction (DAR).
**dar-shift-method** Algorithm to shift the image planes of the cubes according the DAR correction.

0  cpl_image_warp function using the kernel profile TANH
1  cpl_image_warp function using the kernel profile SINC
2  cpl_image_warp function using the kernel profile SINC2
3  cpl_image_warp function using the kernel profile LANCZOS
4  cpl_image_warp function using the kernel profile HAMMING
5  cpl_image_warp function using the kernel profile HANN
6  cpl_image_warp function using the kernel profile NEAREST
7  GSL[?] 2D interpolation function using the interpolation type bilinear
8  GSL[?] 2D interpolation function using the interpolation type bicubic

**dar-shift-length** Kernel length for DAR shift image interpolation, only used if **dar-shift-method** is 0 ... 6. The default value of 0 selects the CPL default kernel length CPL_KERNEL_DEF_SAMPLES which is 2001

**dar-shift-radius** Kernel radius for DAR shift image interpolation, only used if **dar-shift-method** is 0 ... 6. The default value of 0 selects the CPL default kernel width CPL_KERNEL_DEF_WIDTH which is 2.0

**skip_sky_oh_align** The setting skip_sky_oh_align=TRUE applies for a special processing mode for the sky-tweak=1. For the sky-tweak=1 sky subtraction the sky cube can be aligned to the object cube so the wavecal alignment using the OH lines is not needed and may be skipped. But then the "stretch" parameters have to be set. Definitely this needs to be documented in the user manual.

**cube.combine** This parameter allow to de/activate combination of 3D data cubes in case of multi-pointing observations. It maybe used to speed-up data reduction if the user is not interest to the combined cube. If set to false the extracted spectrum will not be created.

**aj-method** When the target is compact, it can be dithered inside the SPIFFER FoV, without the need for sky frames. In this case, the sky in each frame could be estimated by using the mean or the median between the frames, or (probably better), one of the frames closest in time. A negative image of the target will be present in the final data, but this is not a problem if the dithering pattern is accurately designed. This parameter, in case the input data do not contain a dedicated sky observation, controls the creation of a "fake" sky frame from the object frame data. The pipeline currently support the following integer values:

- 0: no sky subtraction.
- 1: sky is taken from the next in MJD-OBS object frame.
- 2: sky is taken from the median of all input sky frames.
- 3: sky is taken from the mean of all input sky frames.
- 4: sky is modeled on each cube plane from the median (or mean) on a user defined rectangular window.
- 5: sky is modeled on each cube plane from the median (or mean) of the medians (or means) computed on four windows taken at the edges of the object FOV.
- 6: sky is modeled on each cube plane from the median (or mean) of the medians (or means) computed on eight windows taken at the border of the object FOV.

- 7: sky is modeled on each cube plane from the median (or mean) computed on the pixels of a region of pixels obtained after `sky-est-niter` iterations where in each are kappa sigma clipped pixels above the median (or mean).

- 8: sky is modeled on each cube plane from the median (or mean) computed on the pixels of a region of pixels obtained using 25% pixels with minimum intensity.

The current pipeline generates 3D data cube that have planes filled by NANs at the start and at the end of the Z axis (wavelength direction). This is a consequence of the brick-wall pattern in the data. For this reason we decided, in case of `aj-method` values between 4 and 8, to skip the first and last 75 planes of the 3D data cube during sky modelling and correction. This affects also the extracted spectrum. We considered this as a negligible effect considering that anyway those planes are fully or partially affected by NAN values, and thus of little use.

The default value is 7 is expected to better estimate the sky in all cases. Method 1 allows to obtain best results if the user plans a proper observation sequence. Results on possible observations are shown in Figures. 12.1. We emphasize that to benefit from this method and observation choice it is up to the user to choose the proper preoptics (we recommend 100mas or 250 mas) and jitter sequence so that every frame is shifted with respect to the two adjacent in MJD-OBS by at least three times the object FWHMs, so that the next frame in MJD-OBS can be a good sky estimation, without overlapping signal.
Figure 12.1: Here we show examples of results obtained with different values of the parameter \( aj \)-method for observations of the quasar J1502+0257 in H band (top), and the object TWA 25, in K band (bottom), corresponding to a plane at a strong sky line (left) and the corresponding extracted spectra (default extraction, mask\_method max, and resampling method DRIZZLE). As we can see for H band without sky subtraction (upper left of shown image panel) the object is barely visible. In all other cases the object can be well detected, and better results are achieved by \( aj \)-method=1. This appears also from the corresponding spectra: in H band (displayed in lin-log scale) in case of no sky subtraction the sky emission lines dominate the continuum, while for other methods are well corrected, better in case of \( aj \)-method=1. In case of K band (displayed in lin-lin scale), results are similar with the difference that in this band one notice the strong thermal emission. Again \( aj \)-method=1 gives better results.

12.8.5 Quality Control Parameters

- QC.LAMBDA.SHIFT.UM [um] OH based center lambda shift
- QC.LAMBDA.SHIFTPIXEL [px] OH based center lambda shift

12.9 eris_ifu_combine_hdr1

12.9.1 Description

The \texttt{eris\_ifu\_combine\_hdrl} recipe align and combine SPIFFIER data cubes using the HDRL resampling routine. By default, the routine will use the WCS in each input cube's header to combine the different cubes. Alternatively, users can provide their own offset list in a two-column ascii file and set offset\_mode=FALSE for the combination, ignoring the WCS expect for referencing the first OBJECT\_CUBE.
12.9.2 Input Frames

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT_CUBE</td>
<td>pro</td>
<td>1 ... n</td>
<td>Science cube(s) to be combined. They should be produced by eris_ifu_jitter recipe; could be OBJECT_CUBE, SKY_TWEAKED_CUBE or DAR_CORRECTED_CUBE</td>
</tr>
</tbody>
</table>

12.9.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Product Depth</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eris_ifu_combine_hdrl_obj_cube_coadd.fits</td>
<td>OBJECT_CUBE_COADD</td>
<td>0</td>
<td>Co-added object cube</td>
</tr>
<tr>
<td>eris_ifu_combine_hdrl_obj_cube_exposure_map.fits</td>
<td>EXPOSURE_MAP</td>
<td>0</td>
<td>Exposure map</td>
</tr>
<tr>
<td>eris_ifu_combine_hdrl_obj_cube_mean.fits</td>
<td>OBJECT_CUBE_MEAN</td>
<td>0</td>
<td>Collapsed object cube</td>
</tr>
</tbody>
</table>

eris_ifu_combine_hdrl_obj_cube_coadd.fits (OBJECT_CUBE_COADD) Combined cube using all OBJECT_CUBE cubes as input.

eris_ifu_combine_hdrl_obj_cube_exposure_map.fits image of the exposure time used on each pixel of the resampled cube.

eris_ifu_combine_hdrl_obj_cube_mean.fits (OBJECT_CUBE_MEAN) Image of the collapsed OBJECT_CUBE_COADD cube calculating the average of each spaxel.

12.9.4 Recipe Parameters

Please see also the description for common recipe parameters in section 12.1 (page 50)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset_mode</td>
<td>bool</td>
<td>False</td>
<td>Offset conventions. If TRUE: the WCS in the input cubes will be used. If FALSE: take user offsets.</td>
</tr>
</tbody>
</table>

continued on next page
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name_i</td>
<td>string</td>
<td>offset.list</td>
<td>Input filename. This must be provided if offset_mode is set to FALSE. It allows the user to set X and Y offsets in a two column format; the offsets must be sorted in the same order as the OBJECT_CUBE files in the sof file. The reference point is the center of the first OBJECT_CUBE in the sof file.</td>
</tr>
<tr>
<td>offset_unit</td>
<td>string enum</td>
<td>PIXEL</td>
<td>Offset unit. &lt;PIXEL</td>
</tr>
<tr>
<td>method</td>
<td>string enum</td>
<td>DRIZZLE</td>
<td>Resampling method</td>
</tr>
<tr>
<td>method.loop-distance</td>
<td>int</td>
<td>3</td>
<td>Loop distance used by all (but NEAREST) methods to control the number of surrounding voxels that are taken into account. A small value allow faster re-sampling but may not give good quality</td>
</tr>
<tr>
<td>method.use-errorweights</td>
<td>bool</td>
<td>False</td>
<td>Use additional weights of $1/err^2$</td>
</tr>
<tr>
<td>method.renka.critical-radius</td>
<td>double</td>
<td>1.25</td>
<td>Critical radius of the Renka method</td>
</tr>
<tr>
<td>method.lanczos.kernel-size</td>
<td>int</td>
<td>2</td>
<td>Kernel size of the Lanczos method</td>
</tr>
<tr>
<td>method.drizzle.downscale-x</td>
<td>double</td>
<td>0.8</td>
<td>Drizzle down-scaling factor in x direction</td>
</tr>
<tr>
<td>method.drizzle.downscale-y</td>
<td>double</td>
<td>0.8</td>
<td>Drizzle down-scaling factor in y direction</td>
</tr>
<tr>
<td>method.drizzle.downscale-z</td>
<td>double</td>
<td>0.8</td>
<td>Drizzle down-scaling factor in wavelength direction</td>
</tr>
<tr>
<td>subtract-background</td>
<td>bool</td>
<td>False</td>
<td>Subtract the background calculated from the median of the 2D image in each channel</td>
</tr>
<tr>
<td>fieldmargin</td>
<td>double</td>
<td>5</td>
<td>Add this margin/border (in percent) to the resampled image/cube</td>
</tr>
<tr>
<td>edge-trim</td>
<td>int</td>
<td>0</td>
<td>Number or pixels to trim for each plane of the input frames. It should be smaller than half image size</td>
</tr>
<tr>
<td>max-cubes-centres-dist</td>
<td>int</td>
<td>240</td>
<td>Maximum distance between cube centers to build a mosaic. Mosaic creation requires a lot of RAM. Users may trim this value to fit RAM resources.</td>
</tr>
</tbody>
</table>

**12.9.5 Quality Control Parameters**

QC.LAMBDA.SHIFT.UM [um] OH based center lambda shift
QC.LAMBDA.SHIFT.PIXEL [px] OH based center lambda shift
12.10 eris_ifu_combine

12.10.1 Description

The eris_ifu_combine recipe align and combine SPIFFIER data cubes using the resampling routine used also in the SINFONI pipeline. By default, the routine will use the WCS in each input cube’s header to combine the different cubes. Alternatively, users can provide their own offset list in a two-column ascii file and set offset_mode=False for the combination, ignoring the WCS expect for referencing the first OBJECT_CUBE. Comparing to eris_ifu_combine_hdrl this recipe provides the ability to apply an initial percentile clipping and a consecutive kappa-sigma-clipping (see parameters \texttt{pclip}, \texttt{ks\_clip}, \texttt{kappa})

12.10.2 Input Frames

<table>
<thead>
<tr>
<th>Frame Tag</th>
<th>Type</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT_CUBE</td>
<td>pro</td>
<td>1 ... n</td>
<td>Science cube(s) to be combined. They should be produced by eris_ifu_jitter recipe; could be OBJECT_CUBE, SKY_TWEAKED_CUBE or DAR_CORRECTED_CUBE</td>
</tr>
</tbody>
</table>

12.10.3 Product Frames

<table>
<thead>
<tr>
<th>Default File Name</th>
<th>Frame Tag</th>
<th>Product Depth</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>out_cube_coadd.fits</td>
<td>OBJECT_CUBE_COADD</td>
<td>0</td>
<td>Co-added object cube</td>
</tr>
</tbody>
</table>

out_obj_cube_coadd.fits (OBJECT_CUBE_COADD) Combined cube using all OBJECT_CUBE cubes as input.

12.10.4 Recipe Parameters

Please see also the description for common recipe parameters in section 12.1 (page 50)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset_mode</td>
<td>bool</td>
<td>False</td>
<td>Offset conventions. If TRUE: the WCS in the input cubes will be used. If FALSE: take user offsets.</td>
</tr>
</tbody>
</table>

continued on next page
### Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>compute_mode</td>
<td>bool</td>
<td>TRUE</td>
<td>Offset conventions. If TRUE: applies reference offset correction. If FALSE: take user offsets. The reference offset is computed as ((\text{min}_\text{off}+\text{max}_\text{off})/2).</td>
</tr>
<tr>
<td>kernel_type</td>
<td>string</td>
<td>TRUE</td>
<td>Offset conventions. If TRUE: applies reference offset correction. If FALSE: take user offsets. The reference offset is computed as ((\text{min}_\text{off}+\text{max}_\text{off})/2).</td>
</tr>
<tr>
<td>name_i</td>
<td>string</td>
<td>offset.list</td>
<td>Input filename. This must be provided if offset_mode is set to FALSE. It allows the user to set X and Y offsets in a two column format; the offsets must be sorted in the same order as the OBJECT_CUBE files in the sof file. The reference point is the center of the first OBJECT_CUBE in the sof file.</td>
</tr>
<tr>
<td>name_o</td>
<td>string</td>
<td>out_coadd_cube.fits</td>
<td>Output filename.</td>
</tr>
<tr>
<td>ks_clip</td>
<td>bool</td>
<td>FALSE</td>
<td>Kappa sigma clipping.</td>
</tr>
<tr>
<td>pclip</td>
<td>double</td>
<td>70</td>
<td>Apply an initial percentile clipping based on the absolute deviation from the median if ks_clip=TRUE.</td>
</tr>
<tr>
<td>kappa</td>
<td>double</td>
<td>2.0</td>
<td>kappa value for sigma clip.</td>
</tr>
<tr>
<td>offset_unit</td>
<td>string</td>
<td>PIXEL</td>
<td>Offset unit. &lt;PIXEL</td>
</tr>
<tr>
<td>subtract-background</td>
<td>bool</td>
<td>False</td>
<td>Subtract the background calculated from the median of the 2D image in each channel</td>
</tr>
<tr>
<td>edge-trim</td>
<td>int</td>
<td>0</td>
<td>Number or pixels to trim for each plane of the input frames. It should be smaller than half image size</td>
</tr>
<tr>
<td>weights</td>
<td>string</td>
<td></td>
<td>Optional input filename. The user can specify his own weighting scheme when compute_mode=MEAN A one column format is expected.</td>
</tr>
</tbody>
</table>

### 12.10.5 Quality Control Parameters

- QC.LAMBDA.SHIFT.UM [um] OH based center lambda shift
- QC.LAMBDA.SHIFT.PIXEL [px] OH based center lambda shift
13 Algorithms

13.1 General Algorithms

13.2 Recipes Algorithms

13.2.1 Dark data reduction

The input dark frames are stacked to generate a master dark frame. Then a map of pixels outliers is determined using the High Level Data Reduction Library based algorithms, either on the master dark (method 2D) or using the set of dark frames (method 3D). Then it is determined the master frame read-out noise.

Readout Noise Computation

The CPL function cpl_flx_get_noise_window is used to compute the readout noise. This function computes the noise in a frame with MonteCarlo approach.

The provided zone is an array of four integers specifying the zone to take into account for the computation. The integers are specified with the recipe parameters qc_{ron/fpn}\_\{x/y\}\{min/max\} where these coordinates are given in the FITS notation (x from 1 to 2048, y from 1 to 2048).

The algorithm will create qc\_{ron/fpn}\_nsamp windows (with the half size qc\_{ron/fpn}\_hsize) on the frame, scattered optimally using a Poisson law. In each window, the standard deviation of all pixels in the window is computed and this value is stored. The readout noise is the median of all computed standard deviations, and the error is the standard deviation of the standard deviations.

13.2.2 Linearity data reduction

A set of input flat (lamp on and lamp off) frames obtained with increasing DET SEQ1 DIT is analysed to determine a map of detector pixels that have a non linear response. The off-frame corrected flat images with intensity below the pixel_saturation level are put in a list of images and a polynomial fit of degree degree is done to each pixel. Fits with a p-value below the threshold set by the parameter p-value are considered bad pixels. Moreover are considered bad pixels also pixels with value smaller than mean - rel-threshold, where the rel-threshold value is set by the parameter rel-chi-low, or greater than mean + rel-threshold, where the rel-threshold value is set by the parameter rel-chi-high. Moreover are considered bad pixels also pixels with value smaller than mean - rel-threshold, where the rel-threshold value is set by the parameter rel-coeff-low, or greater than mean + rel-threshold, where the rel-threshold value is set by the parameter rel-coeff-high.

13.2.3 North-South (distortion) data reduction

The raw detector image is bent in the x-direction. The North-South recipe generates 32 2D-polynomials (one for each slitlet) to correct for this distortion. Any curvature of the detector in y-direction will be corrected by the wavelength calibration.
To measure distortion, the detector is illuminated through a slit mask (three slits for the 250 and 100 mas scales, one slit for the 25 mas scale), which produces vertical traces on the image. In addition, exposures are taken from the arc lamps. Due to the brick wall pattern of the slits, the arc lines can be used to detect the edges of the slits.

With the distortion correction every pixel will warped to a 64 pixel wide image stripe. The left and right edges of the slitlet (taken from the arc lines) are placed at pixel positions 1 and 64. The vertical traces from the slit mask provide additional reference points in the image. The measured points in the detector image are used to fit a 2D-polynomial to obtain the nominal positions in the corrected image strip.

The final product of the north-south (distortion) data reduction is a FITS file containing the 32 2D-polynomials (in 32 FITS extensions) and an extension specifying the left and right most pixels to extract from the raw detector image.

### 13.2.4 Flat data reduction

This recipe determines a master flat frame and a map of pixels with very low response (dead pixels).

### 13.2.5 Wavelength calibration

The wavelength calibration involves the following data reduction steps:

- The input arc line images (WAVE_LAMP, both illuminated and off-lamp frames) are loaded and information on their lamp status (on or off), observed band, preoptic, detector DIT and filename are stored in a table. During this process also a consistency check is performed to make sure frames are of the same band, preoptic and DIT settings. Note that the input frames, usually three taken with the an arc lamp switched on, are obtained with different arc lamps: one with the Argon lamp switched on, one with the Kripton lamp, one with the Neon lamp.

- A reference line list (REF_LINE_ARC) is loaded.

- The optional input master flat (MASTER_FLAT) is loaded.

- If a master flat frame was found the on-off arc frames are divided by the flat.

- The input distortion frame is loaded and using its polynomial distortion coefficients a corrections of the distortion is performed on the arcline frames.

- The slitlets edges position are determined.

- For each IFU slitlet present on the detector a spectrum is determined collapsing the signal along the slitlet extension.

- The arcline positions are determined in a two steps process:
  - first only a few very well isolated arc lines are used to cross correlated them with the ones detected on each of the input arc line frames. Only four or five lines are needed for this first step and the recipes makes sure that both end of the spectra are well covered. The information obtained with the different frames is combined.
– Once these lines are detected a first approximation of the fitting polynomials (FIRST_WAVE_FIT) can be used to perform a second step where a more accurate fit is performed to all detected arclines from the input line list. A wavelength map (WAVE_MAP), an image that stores for each pixel the corresponding wavelength, is determined.

• For quality control the frame is resampled to remove the brick-wall pattern. The corresponding image is an additional quality control product (WAVE_LAMP_RESAMPLED).

13.2.6 OH based Wavelength calibration

To reduce spectral distortions due instrument flexures OH sky lines will be used to recalibrate the wavelength solution.

Since science exposures will typically have integration times of at least a few minutes, the OH sky lines will be bright and clear in individual frames. The processing reconstructs an initial cube from each science frame using the wavelength solution derived from the arc lamp. Then measure the wavelength offset of the frame by comparing the observed wavelengths of the OH lines with respect to their theoretical wavelengths. This offset is folded back into the wavelength solution and the cube reconstructed anew from the raw data and the initial reconstruction is deleted. Thus correcting the spectral flexure does not compromise the quality of the data as does not require additional interpolation steps.

13.2.7 Creation of a 3D data cube

The algorithm described below is executed by recipe eris_ifu_jitter and eris_ifu_stdstar. Given a source image and a corresponding wavelength calibration file, an image is produced in which elements in a given row are associated with a single wavelength. In this way the wavelength shifts between adjacent elements in the rows of the input image are corrected. The output image in the wavelength domain is larger than the input image. Due to the brick wall pattern of the raw frames, some pixels in the first and last few rows have undefined values that are flagged by setting them to NaN. The distribution of these undefined values varies from column to column. The input image is resampled at discrete wavelength intervals using a polynomial interpolation of degree n_coeffs. Different values of the wavelength sampling size and the central wavelength are used for each observed band. Thus, each row has a defined wavelength for each observed band. Since each frame row is now associated with a defined wavelength, each row is used to construct an image which has a defined wavelength. This is done by stacking 32 slitlets on each other, of which each consists of 64 spatial pixels (called also spaxels). Due to the fact that the slitlets length is not exactly 64 pixels and the distance between slitlets is not exactly 64 pixels the edge positions of the slitlets must be known to sub-pixel accuracy. Furthermore, the slitlets must be sorted in the correct sequence to get the correct sequence of the rows in the final images. The centers of each slitlet on the resampled image are determined by averaging the edge positions of the slitlets computed to get the center positions. Then the centers of the slitlets on the raw image are adjusted on the centers of the corresponding ravs of the stacked data cube images. Since only integer pixels can be used a sub-pixel error is made at centering, which is stored and used to shift the rows in the reconstructed images of the data cube to the correct sub-pixel position using a user definable method (objnod-fine_tune_mtd). As the edges of the left-most or right-most slitlets may be too near to or fall outside the image margins, then the fitting of the edges may fail. Then the slitlets distances determined must be used to accurately align each slitlet in the final reconstructed image plane.
13.2.8 Correction of the Differential Atmospheric Refraction

The ERIS-SPIFFIER data reduction pipeline applies a correction of the effects of the atmospheric refraction using the High Level Data Reduction Library. The remind here the main features of the implemented algorithm.

The effects of atmospheric refraction can be readily seen in any spectrograph or IFU. Atmospheric refraction will displace a source, or its spectrum, by an amount that is dependent on the source wavelength and the angular distance of the source from the zenith. This effect is due to the stratified density structure of our atmosphere, and the displacement will be toward the zenith and will be largest for shorter wavelengths. Because of this latter attribute, differential atmospheric refraction is not generally associated with infrared instruments. However, it can be readily seen in ERIS-SPIFFIER data cubes observed with the largest wavelength coverage (H and K bands) and in the smallest pixel scales (25 mas).

This module uses an analytical approach to compute the expected differential refraction as a function of wavelength, zenith angle, and the refractive index of air which, in turn, depends on temperature, pressure, and water vapour pressure.

The hdrl_dar routines require the following inputs (all of which are, generally, available in the input data headers):

- the ambient atmospheric parameters: temperature, pressure, and humidity as contained in the environmental keyword headers: TEL.AMBI.TEMP, TEL.AMBI.PRES.START/END, and TEL.AMBI.RHUM, respectively.
- the instrument rotation angle on the sky
- the parallactic angle of instrument
- and, the world-coordinate system (WCS)

With this input, the differential atmospheric refraction is calculated (optionally, also with an error propagation), and provides an output of the X and Y-axis shifts as a function of the cpl_vector with the input wavelengths. The resulting shift corrections can be directly apply to the pixel image in order to correct this effect. The next section describes the algorithms used to make this calculation.

The High Level Data Reduction Library functions contains routines to calculate the refractive index of air. The main loop compute the differential refractive offset for the input reference wavelengths, and stores the shift in the coordinates, taking into account the instrument rotation angle on the sky and the parallactic angle at the time of observation.

The differential atmospheric refraction is calculated according to the algorithm from Filippenko (1982, PASP, 94, 715). This algorithm uses the Owens formula which converts relative humidity in water vapour pressure.

Owens saturation pressure This function computes the saturation pressure using the J.C. Owens calibration (1967, Applied Optics, 6, 51-59). The saturation pressure is given by:

\[ P_s = \frac{1}{2.303} \ln \left( \frac{H}{H_0} \right) \]

\[ H_0 = 1013.25 \text{ hPa} \]

5See: https://emtoolbox.nist.gov/Wavelength/Documentation.asp#Appendixa, for the formulae used.
where \( \text{verb}+T+ \) is the temperature in Kelvins.

Filippenko refractive index  At sea level (\( P=760 \text{ mm Hg} \), \( T = 15 \text{ C} \)) the refractive index of dry air is given by (Edlén 1953; Coleman, Bozman, and Meggers 1960):

\[
(n(\lambda)_{15,760} - 1)10^6 = 64.328 + \frac{29498.1}{146 - (1/\lambda)^2} + \frac{255.4}{41 - (1/\lambda)^2}
\]

(2)

where \( \lambda \) is the wavelength of light in vacuum (microns). Since observatories are usually located at high altitudes, the index of refraction must be corrected for the lower ambient temperature and pressure (\( P \)) (Barrell 1951):

\[
(n(\lambda)|_{T,P} - 1) = (n(\lambda)_{15,760} - 1) \cdot \frac{P[1 + (1.049 - 0.0157 T)10^{-6} P]}{720.883(1 + 0.003661 T)}
\]

(3)

In addition, the presence of water vapour in the atmosphere reduces \( (n - 1)10^6 \) by:

\[
\frac{0.0624 - 0.000680/\lambda^2}{1 + 0.003661 T} f
\]

(4)

where \( f \) is the water vapour pressure in mm of Hg, and \( T \) is the air temperature in \( \text{C} \) (Barrell 1951) and is expressed as:

\[
f = 0.75006158 \cdot s_p \cdot h
\]

(5)

where \( s_p \) is the saturation pressure of equation 1 and \( h \) is the relative humidity in \( [\%] \).

13.2.9 Sky model

As part of its data processing procedures, the SPIFFIER DRS make use of a sky subtraction technique which is detailed in Davies (2007) [RD3]. The method was developed for SINFONI data but is equally applicable to any other near infrared data cubes. It is also implemented in the VLT KMOS pipeline (see [RD9] section 7.2.4 and page 97/98). It makes use of the fact that most of the variation on the 1-2.5 \( \mu \text{m} \) OH line ratios is due to variations on the vibrational temperature of the OH radical. Fortuitously, to a reasonable approximation, lines from the same vibrational transitions all lie within a well defined wavelength region. Hence one can apply a single scaling across the whole of this region, which is therefore robust against biasing if line emission from the science object occurs at the same wavelength as an OH line. A basic correction can also be made for changes in the rotational temperature, although in this case each spectral segment comprises several short discrete regions. Once the vibrational variations are corrected, any rotational variations can easily be measured and also corrected. While no special mathematical manipulations are required, this procedure is included here because it is not considered a “standard” technique.

1. Identify spaxels with least flux in object cube.
2. Sum spectra from these spaxels in both object and sky cubes separately.

3. Fit a blackbody function to the underlying continuum in the sky spectrum (the thermal background).

4. The fitted function is subtracted from both the original object and sky cubes and from the extracted object and sky spectra.

5. The spectra with removed thermal background are compared with regard to offsets in bright OH lines. The sky cube (with removed thermal background) is shifted accordingly. Note that for SPIFFIER, the default is for spectral flexure to already be corrected. However there may be some situations where this is not so, in which case this step is carried out here.

6. Again the spectrum of the processed object and sky cubes are extracted using the same mask as in step 1.

7. To correct vibrational variations, the spectra are divided into segments along the wavelength axis. For each segment the spectral vectors of bright OH lines are extracted.

8. The sky spectrum is scaled to match the object spectrum in each spectral segment.

9. The scalings of each spectral segment are combined to a single scaling function which is applied to the sky spectrum.

10. To correct rotational variations, steps 7 to 9 are repeated.

11. The two scaling functions are multiplied.

12. The resulting scaling function is multiplied with the compensated sky cube which in turn is subtracted from the compensated object cube.

### 13.2.10 Cube resampling

The ERIS-SPIFFIER pipeline uses the High Level Data Reduction Library based algorithm to resample 3D data cubes. We remind here the main features.

The resampling High Level Data Reduction Library project addresses two different scenarios. Assuming that one would like to re-grid and stack 10 dithered images/cubes with pre-determined World Coordinate System (WCS). Then one should be able to:

- Combine all images/cubes in a single cloud of points (we use a table for this step - see below) and when creating the resampled output image/cube all information is used in a single interpolation step, i.e. no image-by-image interpolation is done.

- Resample each individual image/cube on the same output grid (image by image interpolation) and combine the single resampled images in a second step with hdrl stacking methods.

Both scenarios interpolate only once, but the second scenario is important in case not all bad pixels could be properly determined and inserted in the bad pixel mask upfront. Then the implemented HDRL pixel resampling
function offers the possibility to use a robust pixel combination method (median or $\kappa\sigma$ clipping) to combine the images/cubes.

Currently there are six different interpolation methods provided by High Level Data Reduction Library:

- **Nearest**: Nearest resampling
- **Linear**: Weighted resampling using an inverse distance weighting function
  - The parameter `loop_distance` controls the number of surrounding pixels that are taken into account on the final grid, e.g. a `loop_distance` of 1 uses 3 pixels ($x - 1$, $x$, $x + 1$) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter `use_errorweights` is set to TRUE, an additional weight, defined as 1/variance, is taken into account.
- **Quadratic**: Weighted resampling using a quadratic inverse distance weighting function
  - The parameter `loop_distance` controls the number of surrounding pixels that are taken into account on the final grid, e.g. a `loop_distance` of 1 uses 3 pixels ($x - 1$, $x$, $x + 1$) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter `use_errorweights` is set to TRUE, an additional weight, defined as 1/variance, is taken into account.
- **Renka**: Weighted resampling using a Renka weighting function
  - The parameter `critical_radius` defines the distance beyond which the weights are set to 0 and the pixels are therefore not taken into account. The parameter `loop_distance` controls the number of surrounding pixels that are taken into account on the final grid, e.g. a `loop_distance` of 1 uses 3 pixels ($x - 1$, $x$, $x + 1$) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter `use_errorweights` is set to TRUE, an additional weight, defined as 1/variance, is taken into account.
- **Drizzle**: Weighted resampling using a drizzle-like weighting scheme
- **Lanczos**: Weighted resampling using a lanczos-like restricted sinc as weighting function

**Nearest Neighbour** The algorithm does not use any weighting function but simply uses the value of the nearest neighbour inside an output voxel centre as the final output value. If there is no nearest neighbour inside the voxel (but e.g. only outside), the voxel is marked as bad. This speeds up the algorithm considerably. There are no control parameter for this method.

**Linear** The algorithm uses a linear inverse distance weighting function ($\frac{1}{r}$) for the interpolation. The parameter `loop_distance` controls the number of surrounding pixels that are taken into account on the final grid, e.g. a `loop_distance` of 1 uses 3 pixels ($x - 1$, $x$, $x + 1$) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter `use_errorweights` is set to TRUE, an additional weight, defined as 1/variance, is taken into account.

**Quadratic** The algorithm uses a quadratic inverse distance weighting function ($\frac{1}{r^2}$) for the interpolation. The parameter `loop_distance` controls the number of surrounding pixels that are taken into account on the final grid, e.g. a `loop_distance` of 1 uses 3 pixels ($x - 1$, $x$, $x + 1$) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter `use_errorweights` is set to TRUE, an additional weight, defined as 1/variance, is taken into account.

**Renka** The algorithm uses a modified Shepard-like distance weighting function following Renka for the interpolation. The parameter `critical_radius` defines the distance beyond which the weights are set to 0 and the pixels are therefore not taken into account. The parameter `loop_distance` controls the number of surrounding pixels that are taken into account on the final grid, e.g. a `loop_distance` of 1 uses 3 pixels ($x - 1$, $x$, $x + 1$) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter `use_errorweights` is set to TRUE, an additional weight, defined as 1/variance, is taken into account.

---

**Drizzle**  The algorithm uses a drizzle-like distance weighting function for the interpolation. The down-scaling factors\( \text{pix\_frac}_x, \text{pix\_frac}_y, \text{and pix\_frac}_\lambda \), for \( x, y, \) and wavelength direction control the percentage of flux of the original pixel/voxel that drizzles into the target pixel/voxel. The parameter\( \text{loop\_distance} \) controls the number of surrounding pixels that are taken into account on the final grid, e.g. a \( \text{loop\_distance} \) of 1 uses 3 pixels \((x - 1, x, x + 1)\) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter\( \text{use\_errorweights} \) is set to TRUE, an additional weight defined as \( 1/\text{variance} \) is taken into account\(^7\).

**Lanczos**  The algorithm uses a restricted \( \text{sinc} \) distance weighting function \( (\frac{\text{sinc}(r)}{\text{sinc}(r/k_{\text{size}})}) \) with the kernel size given by the parameter\( \text{kernel\_size} \) for the interpolation. The parameter\( \text{loop\_distance} \) controls the number of surrounding pixels that are taken into account on the final grid, e.g. a \( \text{loop\_distance} \) of 1 uses 3 pixels \((x - 1, x, x + 1)\) in each dimension, i.e. 9 in total for a 2D image and 27 in total for a 3D cube. Moreover, if the parameter\( \text{use\_errorweights} \) is set to TRUE, an additional weight defined as \( 1/\text{variance} \) is taken into account\(^7\).

Note that because High Level Data Reduction Library propagates the full data, error and pixel quality information and to perform the resampling step in a general way need to fill an intermediate pixel table with the pixel data, error, pixel quality and corresponding WCS information, this step is demanding in terms of RAM. We could reduce most of the commissioning and early operation data on a computer with 32 GB of RAM. In a few cases, if the user maps a large portion of the FOV, and thus the output cube mosaic may be very large, one may require even more.

Please note that the output is using the Gnomonic projection\(^8\). The gnomonic projection is from the centre of a sphere to a plane tangential to the sphere. The sphere and the plane touch at the tangent point. This is encoded in the header file by:

\[
\begin{align*}
\text{CTYPE1} &= \text{’RA--TAN’} / \text{Gnomonic projection} \\
\text{CTYPE2} &= \text{’DEC--TAN’} / \text{Gnomonic projection}
\end{align*}
\]

- Combine all images/cubes in a single cloud of points (we use a table for this step - see below) and when creating the resampled output image/cube all information is used in a single interpolation step, i.e. no image-by-image interpolation is done.
- Resample each individual image/cube on the same output grid (image by image interpolation) and combine the single resampled images in a second step with hdrl stacking methods.

Both scenarios interpolate only once, but the second scenario is important in case not all bad pixels could be properly determined and inserted in the bad pixel mask upfront. Then the implemented HDRL pixel resampling function offers the possibility to use a robust pixel combination method (median or \( \kappa \sigma \) clipping) to combine the images/cubes.

### 13.2.11 Instrument Response computation

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

\(^8\)see [https://en.wikipedia.org/wiki/Gnomonic_projection](https://en.wikipedia.org/wiki/Gnomonic_projection)
velocity as the observed spectrum and then interpolate the reference spectrum to the same steps as the observed spectrum. We apply a correction of the telluric absorption by looking for the best fitting spectrum within a catalogue of telluric model spectra. The best telluric model is the one that minimises the mean of a “correction” spectrum computed in the following intervals:

This spectrum is obtained via the following steps:

1. adjust telluric model spectrum to wavelength scale of observed spectrum (small shifts may occur due to imperfect wavelength calibration)
2. convolve the telluric spectrum to the same resolution as the observed spectrum
3. divide the observed spectrum by the shifted and convolved telluric spectrum
4. fit the continuum of this ratio at pre-defined wavelength points
5. divide ratio by fit

Since the model spectra are provided for a large range of water vapour content, but only for few values of CO$_2$, we mask CO$_2$ regions when fitting the response later. The response is computed by dividing this telluric-corrected model spectrum of the standard star (in erg cm$^{-2}$s$^{-1}$Å$^{-1}$) by the 1D extracted observed spectrum of the standard star corrected for gain, exposure time, atmospheric extinction (the atmospheric extinction table is interpolated to get the same binning) and for telluric absorption. To reduce the noise of the resulting ratio spectrum we apply a median filter of eleven pixels half width. Then we apply a cubic-spline fit to the points defined in RESP_FIT_POINTS_CATALOG to get the final response. Regions of very high telluric absorption (between J and H and between H and K in the NIR arm) or CO$_2$ absorption (see above) are masked and interpolated.

The response is obtained with the following equation:

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

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

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

\[
\text{erg/s/cm}^2/\text{Å} = \frac{\text{ADU/pixel}[\text{Å}] \times \text{Response}[\text{erg/e}^{-}/\text{cm}^2] \times 10^{(0.4 \cdot \text{airmass/ext})}}{\text{gain[ADU/e]} \times \text{Exptime[s]} \times \text{bin_size}}
\]

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

13.2.13 Strehl computation

The central maximum in units of the total flux of a PSF is a measure of the image quality and AO performance. The Strehl ratio is the observed ratio in units of theoretically possible (diffraction limited) ratio. It is a number usually between \( \approx 0.01 \) (seeing limited) and 0.6-0.8 depending on the performance and the ambient conditions.

As the ERIS-SPIFFIER possible FOV are either too narrow or too large for most of the calibration stars used to measure the instrument Strehl, it has been defined a special template in which the same PSF standard is observed first in the 0.025 mas scale, to accurately determine the object image Point Spread Function maximum, and the with the 0.100 mas scale to determine the star flux and sky background levels.

Normally the Strehl is determined as ratio:

\[
\text{Strehl} = \frac{l_{\text{max}}/\text{flux}}{PSF_{\text{max}}/\text{flux}_{\text{PSF}}}
\]

Where \( l_{\text{max}} \) is the maximum intensity of the object (sky background subtracted), and \( \text{flux} \) the corresponding flux, while \( PSF_{\text{max}} \) is the maximum intensity of a corresponding theoretical PSF corresponding to a telescope with the same diameters for the primary and secondary mirrors, at a given wavelength and in a given wavelength range, at a given pixel scale, with a given size.

This formula is applied on both images of the PSF standard taken with two camera pixel scales (25 and 100 mas).

The ERIS-SPIFFIER pipeline implements the same algorithm as in the Paranal Multi Strehl Meter GUI.

Initially it is performed a 2D Gaussian fit to the image resulting averaging the STD star cube along the wavelength (z) axis. This fit determines the approximate position of the STD star image, the FWHM along the major and minor axes and the orientation of the elliptic isophotes. Then the sky background is subtracted by taking the median value of the four image plane’s corner regions, \( 8 \times 8 \) pixels each. Finally it is determined the maximum of the resulting sky corrected image.

The star flux is defined as the flux of the STD star above the sky background.

Then it is built the instrument PSF by taking into account also of the possible anamorphysm and occultation introduced by the secondary mirror and the PSF’s maximum and flux are determined.
13.2.14 Standard star position detection

The recipe eris_rec_jitter determines the position of the maximum in the image obtained by collapsing the cube along the z-axis. Then the STD star object (which is assumed to be the only object in the FOV) is more accurately located using a centroid algorithm assuming a 2D Gaussian shape approximation for the object PSF, which also allows to estimate the STD star FWHM along the X and Y directions.
A  Installation

A.1  System Requirements

The ERIS-SPIFFIER pipeline implements recipes and algorithms in a very efficient manner. The only system
recommendation we would like to give to the user, as the implementation employs OpenMP instructions, is
to run the recipe on a system with multi core architecture and enough RAM and disk space. For example the
following requirements are recommended:

• 20 (minimum)/(32 or more is better) GB of memory
• 4 (more is better) CPU cores (physical cores)
• 1 TB of free disk space, possibly SSD.
• GCC 8.3.1 (or newer)

In this release we have trimmed the default of the parameter max-cubes-centres-dist of the jitter and stdstar
recipes to a value (240) that should allow also users with 20 GB of RAM to run that recipe, eventually not
generating some products in case the maximum distance between the centres of the 3D cubes to be combined
is greater than that value. However we recommend user to have more RAM.

A.2  Installing the ERIS Pipeline

Installation via RPM or MacPorts is recommended. See Installation instructions for all the details.

In case the user platform is not one for which an RPM or MacPorts are provided the user may use the
install_esoreflex script.

ESOReflex can be installed as:

./install_esoreflex

Then follow the instructions on the screen (selecting ERIS). Once the script finishes successfully and the path
variables have been set, the installation of the ERIS pipeline ESOReflex workflow and the pipeline are complete.

A.2.1  Build Requirements

There are no particular build requirements for the ERIS pipeline. Installation based on the install_esoreflex
script, requires the user has installed on her/his desktop certain software as describe at: