



EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

ESO - European Southern Observatory
Karl-Schwarzschild Str. 2, D-85748 Garching bei München

Very Large Telescope Paranal Science Operations CRIRES data reduction cookbook

Doc. No. VLT-MAN-ESO-14200-4032

Issue 93.2, Date 22/12/2013

J. Smoker, E.Valenti, D. Asmus,
P. Birstow, A. Smette, M. Hilker, B. Wolff, Y. Jung
Prepared
Date Signature

A. Kaufer
Approved
Date Signature

C. Dumas
Released
Date Signature

This page was intentionally left blank

Change Record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
80.0	11/10/2007	all	cookbook creation and first archiving
88.0	01/10/2011	all	major updates
89.0	27/04/2012	all	comments requested
89.1	20/06/2012	all	major updates
89.2	13/07/2012	all	major updates
89.3	18/09/2012	all	updates
89.4	24/09/2012	all	updates
90.0	20/10/2012	all	final updates
91.0	05/11/2012	all	document released
93.1	27/08/2013		Acknowledgments updated
"	27/08/2013	1.1	Typos.
"	27/08/2013	3.3.1	Comment on RMS noise for windowed and normal mode against DIT.
"	27/08/2013	5	Flux cals only taken for 5 standard settings.
"	27/08/2013	7.3	Recall y_pos_N refers to position where the wavelength calibration is applied.
"	27/08/2013	7.3	Use "spec_zone" to extract at a certain "Y" value.
"	27/08/2013	8.1	Application of non-linearity coefficients explained.
"	27/08.2013	8.3	Updated.
"	27/08.2013	8.9	Telluric line section updated.
"	27/08.2013	10.1.2	UNe likely to replace ThAr sometime during P93.
"	27/08/2013	10.4	Non linearity correction section updated.
"	27/08/2013	Appendix B	IRAF procedure to read in reduced files updated.
"	27/08/2013	Appendix C	Csh script to associate science and calibrations and reduce the data using esorex added.
93.2	22/12/2013	3.3.5	Comment on creating DETMON files added.
"	22/12/2013	8.3	"
"	22/12/2013	C	Write to crires@eso.org for simple reduction script.

Acknowledgments

The layout of this cookbook was initially based on the ISAAC data reduction guide written by J.G. Cuby, P. Amico, N. Devillard and C. Lidman. We thank Ronald Mennickent Cid for details of importing reduced CRIRES spectra into IRAF, to Pedro Figuera for turning this into an IRAF procedure and to Ian Crossfield for comments.

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Reference documents	1
1.3	Abbreviations and acronyms	1
1.4	Stylistic conventions	2
1.5	Topics not covered by the cookbook	2
1.6	Suggestions for improvements to the cookbook	2
2	CRIRES FITS Information	3
2.1	Extracting FITS Information	3
2.2	Visualising FITS Tables	4
2.3	File Names	4
2.4	Telescope Keywords	5
2.5	Instrument Keywords	6
2.6	Observation Block Keywords	6
2.7	Template Keywords	7
2.7.1	Keywords common to all templates	7
2.8	Pipeline Products	7
2.9	Summary Table	8
3	CRIRES data and calibrations	10
3.1	Readout modes	10
3.2	EXPTIME, NDIT and DIT	11
3.3	Calibrations	11
3.3.1	Darks	11
3.3.2	Flatfields	11
3.3.3	ThAr wavelength calibration	15
3.3.4	Gas-cell wavelength calibrations	15
3.3.5	Detector linearity images	15
3.4	General Features of CRIRES Infrared spectra	16
3.4.1	Glow	16
3.4.2	Optical ghosts	16
3.4.3	Odd-even column effect and detector non-linearity	16
3.4.4	Remnants from bright sources	16
3.4.5	Fringing	17
4	Downloading CRIRES raw data and calibrations	17
5	Reducing the science data	26
5.1	First subtraction	26

5.2	Flat fielding	26
5.3	Wavelength calibration	27
5.4	Combining 2d spectra	27
5.5	Extraction	28
5.6	Removing telluric lines	28
5.7	Flux calibration	29
6	The ESO CRIRES pipeline: installation	30
6.1	Opening gasgano, resizing columns and selecting data for reduction	31
6.2	The ESO CRIRES pipeline under MacOS	32
6.3	The ESO CRIRES pipeline reflex environment	33
7	The ESO CRIRES pipeline: Quick look data reduction of CRIRES nodding and stare images using the recipe crires_spec_jitter	36
7.1	Data reduction of nodding spectra using gasgano	36
7.2	Quick data reduction of nodding spectra using esorex	42
7.3	Quick reduction of stare frames using esorex	42
7.4	Plotting the reduced spectra	44
8	The ESO CRIRES pipeline: Full data reduction of CRIRES nodding images	48
8.1	Look at the raw data and calibrations	48
8.2	Master dark creation	49
8.2.1	Reduction of darks within gasgano	50
8.2.2	Reduction of darks within esorex for flats and science frames	52
8.2.3	Inspection of pipeline results for master dark and QC parameters	52
8.3	Linearity correction using detmon_ir_lg	55
8.4	Master flatfield creation	55
8.4.1	Reduction of flatfields within gasgano	55
8.4.2	Reduction of flatfields within esorex	58
8.4.3	Inspection of pipeline results for master flatfield and QC parameters	59
8.5	Strategies for wavelength calibration	62
8.5.1	Wavelength less than 2.4 microns with more than two or three well-spaced ThAr lines available per detector	62
8.5.2	Wavelength less than 2.4 microns with less than two ThAr fibres available per detector	62
8.5.3	Wavelength more than 2.4 microns with gas-cell spectra available	62
8.5.4	Science frame with no gas cell or ThAr wavelength calibration and no OH or sky absorption lines	63
8.6	Wavelength calibration examples	66
8.6.1	Reduction of ThAr frames using gasgano	66
8.6.2	Reduction of ThAr frames using pipeline 2.1.3 or earlier and esorex	68
8.6.3	Reduction of ThAr frames using pipeline 2.3.1 or later and esorex	73

8.6.4	Reduction of OH line frames using esorex	73
8.6.5	Use of the physical model and esorex to improve the wavelength solution	74
8.6.6	Inspection of the pipeline results produced by crires_spec_wave to check on the wavelength calibration solution	78
8.7	Combining nodding spectra	79
8.7.1	Reduction of nodding frames using gasgano	82
8.7.2	Reduction of nodding frames using esorex	82
8.7.3	Inspection of pipeline results produced by crires_spec_jitter	83
8.8	Reducing CRIRES data semi-automatically	88
8.9	Telluric line correction	88
8.10	Importing reduced data into IDL or IRAF	88
8.10.1	IDL	88
8.10.2	IRAF	88
8.11	Plotting reduced spectra	88
9	Utilities recipes	88
9.1	Examples of the use of the utilities	91
9.1.1	crires_util_combine	91
9.1.2	crires_util_extract	91
9.1.3	crires_util_wlcalib	91
9.1.4	crires_util_wlassign	91
9.1.5	crires_util_genlines	92
10	Limitations and problems	92
10.1	Wavelength calibration issues	92
10.1.1	Limited accuracy in the ThAr and OH line catalogues	92
10.1.2	Too few ThAr lines	92
10.1.3	False identification of ThAr line features	92
10.2	Problems in tracing	92
10.3	IRAF issues	92
10.4	Correcting for non-linearity	95
10.5	Problems in running gasgano	95
10.6	Cannot find recipe files	97
10.7	Problems in running esorex	97
A	Example IDL script to read in reduced data and produce ascii-format output files	98
B	Example IRAF script to read in reduced data and produce ascii-format output files	101
C	Script to associate science and calibration data, create master frames and extract using crires_spec_jitter	103

1 Introduction

1.1 Purpose

The document is intended for astronomers who want to reduce CRIRES data. It describes the various data formats delivered by CRIRES, observational scenarios and reduction procedures. The document also describes the algorithms implemented in the CRIRES pipeline data reduction package. Though the CRIRES pipeline is described in detail in the CRIRES pipeline User Manual (see <http://www.eso.org/pipelines/>), here we will list and present some of the CRIRES pipeline routines. The current document is aimed at describing the algorithmic concepts present in the recipes, and provide help for users who want to analyze data from CRIRES from scratch. That said, the pipeline manual should be considered the main reference.

The pipeline produces reduced data; however, it is not meant to replace more general reduction packages such as IRAF, MIDAS or IDL.

This document does *not* describe the CRIRES instrument, its mode of operations, how to acquire data, the offered templates, or the various issues attached to Phase II Observing Preparation. The reader is assumed to have read the CRIRES User's Manual beforehand, and have a basic knowledge of infrared data reduction in spectroscopy.

1.2 Reference documents

- 1 CRIRES User Manual
- 2 CRIRES Pipeline User Manual

The latest version of the CRIRES pipeline should always be used, this is available at:

<http://www.eso.org/projects/dfs/dfs-shared/web/vlt/vlt-instrument-pipelines.html>

The latest version of the user manual is available here:

<http://www.eso.org/sci/facilities/paranal/instruments/crides/doc/>

1.3 Abbreviations and acronyms

The following abbreviations and acronyms are used in this document:

DCS	Detector Control System
DIT	Detector Integration Time
DPR	Data Products (FITS identifier)
SciOps	Science Operations
ESO	European Southern Observatory
Dec	Declination
eclipse	ESO C Library Image Processing Software Environment
ESO-MIDAS	ESO's Munich Image Data Analysis System
FITS	Flexible Image Transport System
IRAF	Image Reduction and Analysis Facility
PAF	PArameter File
RA	Right Ascension
UT	Unit Telescope
VLT	Very Large Telescope

1.4 Stylistic conventions

The following styles are used:

- bold** in the text, for commands, etc., as they have to be typed.
- italic* for parts that have to be substituted with real content.
- box for buttons to click on.
- `teletype` for examples and filenames with path in the text.

Bold and *italic* are also used to highlight words.

1.5 Topics not covered by the cookbook

The following topics are either not covered, or incompletely covered by the current version of the cookbook:

- Reduction of generic offset observations.
- A full explanation of 2D wavelength calibration. From version 2.3.1 of the pipeline the ThAr pinhole data are now properly supported. The wavelength calibration is applied at the different Y positions.
- Some figures containing gasgano screenshots display still show the old `wl_starty` `wl_stop` parameters that were updated in v2.3.1 of the pipeline.
- Examples of the pipeline extraction at low S/N ratios.
- Examples of how well the linearity correction works at low- and high- S/N ratios.
- Residuals obtained using the CRIRES model at wavelengths shorter than 3.5 microns.

It is hoped that a future version will include these topics.

1.6 Suggestions for improvements to the cookbook

If you have suggestions for improvements to the cookbook or find mistakes in it, please email crires@eso.org. This account is read by the Paranal Instrument Scientist who is currently the author of the cookbook. Questions about the pipeline itself should be sent to usd-help@eso.org, or better by using the "Ask for help" link within the ESO user portal here:

<http://www.eso.org/UserPortal/>.

Note that many of the examples in this cookbook refer to version 2.1.3 of the pipeline, however, the newest version should always be used (which at the time of writing is 2.3.1).

Updates to the cookbook will be announced in the [CRIRES news pages](#).

2 CRIRES FITS Information

Ancillary data attached to CRIRES files are all written into FITS headers. The following chapter lists the most important keywords and describes commands on how to retrieve them. For ease of reading, keywords are shortened from `HIERARCH ESO A B C` to `A.B.C` (shortFITS notation).

Notice that all of this information is present in the ESO dictionaries, available from the ESO archive Web site here:

<http://archive.eso.org/cms/tools-documentation/eso-data-interface-control.html>

The following section only tries to summarise the most important information.

2.1 Extracting FITS Information

There are many tools to extract and parse FITS headers. One convenient way of extracting FITS information and displaying it on a terminal or re-directing it to a text file, is to use two stand-alone programs called **dfits** and **fitsort**, which are both included in the *scisoft* package and also available here:

<http://archive.eso.org/saft/>

dfits dumps a FITS header to stdout. You can use it to dump the FITS headers of many files, to allow the parsing of the output. For example:

```
[jsmoker@pc015042 src]$ dfits
```

```
usage: dfits [-x xtnum] <list of FITS files>
```

```
usage: dfits [-x xtnum] -
```

The former version expects file names.

The latter expects data coming in from stdin.

`-x xtnum` specifies the extension header to print

`-x 0` specifies main header + all extensions

...or in action:

```
% dfits *.fits | grep "TPL ID"
```

Usually, you want to get the value of a list of given FITS keywords in a list of FITS files. **fitsort** reads the output from **dfits**, classifies the keywords into columns, and prints out in a readable format the keyword values and file names. An example follows:

```
% dfits *.fits | fitsort NAXIS1 NAXIS2 BITPIX
```

fitsort also understands the shortFITS notation, where e.g. `HIERARCH ESO TPL ID` is shortened to `TPL.ID`. A classification example could be (both commands are equivalent, since **fitsort** is case-insensitive):

```
dfits CR*fits | fitsort TPL.ID DPR.TYPE INS.WLEN.REF INS.SLIT1.WID AOS.RTC.LOOP.STATE
                                DET.DIT DET.NDIT
dfits CR*fits | fitsort tpl.id dpr.type ins.wlen.ref ins.slit1.wid aos.rtc.loop.state
                                det.dit det.ndit
```

The output from this combination is something like:

```
CRIRES_SPEC_DARK108_0013.fits  CRIRES_spec_cal_Darks          DARK  4092.900  0.200  OPEN
                                2.00000000 3 (all on one line)
CRIRES_SPEC_FLAT108_0001.fits  CRIRES_spec_cal_LampFlats       FLAT  4092.900  0.200  OPEN
                                10.00000000 6 (all on one line)
CRIRES_SPEC_STD076_0011.fits   CRIRES_spec_cal_AutoNodOnSlit  STD   2287.000  0.400  CLOSED
                                20.00000000 4 (all on one line)
CRIRES_SPEC_STD076_0012.fits   CRIRES_spec_cal_AutoNodOnSlit  STD   2287.000  0.400  CLOSED
                                20.00000000 4 (all on one line)
```

...which shows the filename, name of template used to produce the file, data product type, reference wavelength, slit width, loop status (i.e. if the adaptive optics was deployed or not), DIT and NDIT.

This kind of table is useful in getting an idea of what is present in a directory or list of directories. Loading such a summary table into a spreadsheet program also makes it conveniently readable.

Another easy way to visualise the content of a FITS file is using the graphical *Gasgano* tool, which is the data management tool currently used at VLT. *Gasgano* offers an automatic data classification only by applying an instrument specific rule. A view on a set of CRIRES data is shown in Fig. 2.1 as an example, where the keywords `FILE`, `TPL.ID` and `DPR.TYPE` have been specified in the appropriate *Preferences* fields. More information about a single file can be obtained by clicking on its name: the corresponding FITS file header will be displayed on the bottom panel, where specific keywords can be opportunely filtered and searched.

See the *Gasgano User's Manual* for a complete description of the *Gasgano* interface. It is available at <http://www.eso.org/sci/software/gasgano/>. Sections 6.1 and 10.5 also provide some information on the use of *gasgano*.

2.2 Visualising FITS Tables

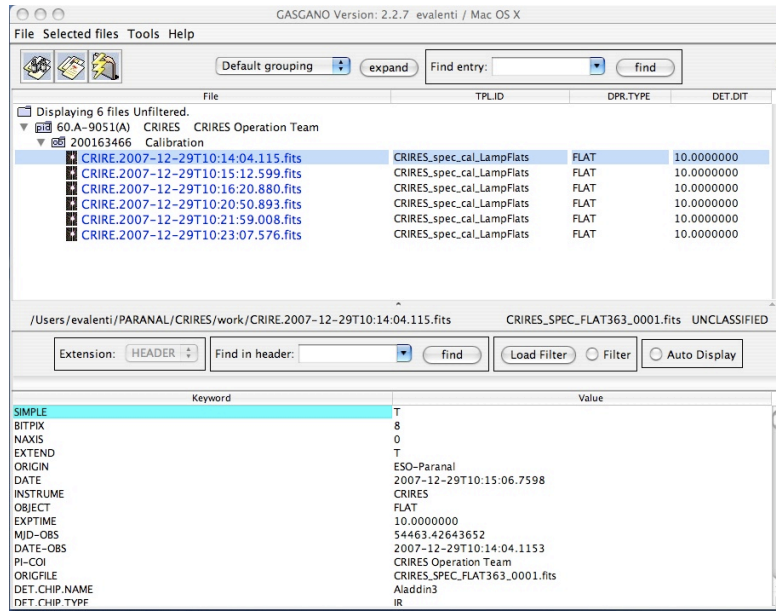
Similarly, a FITS table can be visualised on the command-line without the need to call a full-fledged FITS-capable package. The `dtfits` command has been written for precisely that purpose. You will find it useful for spectroscopic data reduction, if you need to check out the results of the pipeline recipes that produce FITS tables.

`dfits` and `fitsort` will help you classify tables and see ancillary data attached to them, but `dtfits` will display all information contained in the table itself, in ASCII format on the command-line. There are various options to help make the output readable on a terminal, or by a spreadsheet program.

See the `dtfits` manual page to get more information.

2.3 File Names

- `PIPEFILE` (if set) contains the name of the product using the official naming scheme for

Figure 1: *Gasgano* main window.

CRIRES products.

- `ORIGFILE` contains the name of the file on the instrument workstation.
- `ARCFILE` is the archive file name.

2.4 Telescope Keywords

Here is a non-exhaustive list of telescope keywords.

- `RA` Right ascension (J2000) in degrees. Notice that the comment field indicates the value in hh:mm:ss.s format.
- `DEC` Declination (J2000) in degrees. Notice that the comment field indicates the value in hh:mm:ss.s format.
- `INS1.DROT.POSANG` Position angle on sky as measured from North to East (degrees).
- `TEL.AIRM.START` Airmass at start.
- `TEL.AIRM.END` Airmass at end.
- `TEL.AMBI.FWHM.START` Astronomical Site Monitor seeing at start. Note that this value might differ significantly from the CRIRES image quality, which is usually better.
- `TEL.AMBI.FWHM.END` Astronomical Site Monitor seeing at end. Note that this value might differ significantly from the CRIRES image quality, which is usually better.
- `TEL.IA.FWHMLIN` Delivered seeing on Image Analysis detector at ~ 550 -nm.
- `TEL.IA.FWHMLINOBS` Delivered seeing on Image Analysis detector at ~ 550 -nm.

- `TEL.AMBI.IWV.START` Average of the Integrated Water Vapor measurements towards zenith over the previous 2min at start of exposure. Implemented soon. This can be compared with the requested value `ESO.OBS.WATERVAPOUR`.
- `TEL.AMBI.IWSTD.START` Standard deviation of the Integrated Water Vapor measurements towards zenith over the previous 2 min, at start of exposure. Implemented soon.

2.5 Instrument Keywords

Here is a non-exhaustive list of instrument keywords.

- `INS.SLIT1.WID` Entrance slit width (since Oct 2011 set to 0.2 or 0.4 arcseconds).
- `INS.WLEN.REF` Reference wavelength (microns).
- `INS.WLEN.CWLEN` Central wavelength (microns).
- `INS.WLEN.END1` Wavelength at the end of detector 1 (microns).
- `INS.WLEN.END2` Wavelength at the end of detector 2 (microns).
- `INS.WLEN.END3` Wavelength at the end of detector 3 (microns).
- `INS.WLEN.END4` Wavelength at the end of detector 4 (microns).
- `INS.WLEN.MIN` Minimum wavelength of the unvignetted spectral range (microns).
- `INS.WLEN.MAX` Maximum wavelength of the unvignetted spectral range (microns).
- `INS.GRAT.ORDER` Grating order.
- `INS.LAMP1.ST` Halogen lamp status.
- `INS.LAMP2.ST` IR Emitter lamp status.
- `INS.LAMP3.ST` NeKr lamp status.
- `INS.LAMP4.ST` Thorium-Argon lamp status.
- `INS.DROT.MODE` Derotator mode (e.g. SKY or ELEV).
- `DET.DIT` Detector Integration Time (seconds).
- `DET.NDIT` Number of averaged DITs.
- `AOS.RTC.LOOP.STATE` Adaptive optics loop state (closed or open).

2.6 Observation Block Keywords

- `OBS.PROG.ID` Program ID.
- `OBS.NAME` Name of the OB (as prepared with P2PP).
- `OBS.TARG.NAME` Target package name (as prepared with P2PP).

2.7 Template Keywords

2.7.1 Keywords common to all templates

- `TPL.ID` contains an unique identifier describing the template which was used to produce the data.
- `DPR.CATG` Data Product category (SCIENCE, CALIB, ...). Frame selection in the pipeline is mostly based on this keyword value.
- `DPR.TYPE` Data Product type (OBJECT, SKY, ...).
- `DPR.TECH` Data Product acquisition technique (e.g. IMAGE, SPECTRUM-NODDING, SPECTRUM).
- `TPL.NEXP` Number of scheduled exposures within the template. ¹
- `TPL.EXPNO` Exposure rank number within template.

A template may produce several different frame types. Frames are discriminated by the value of the DPR keywords: `DPR.CATG`, `DPR.TYPE`, and `DPR.TECH` take different values depending on the observed frame type.

The offsets sent to the telescope for jitter observations (only relevant if the *Jitter width* is greater than 0) are stored into 4 keywords. This applies to AutoNodOnSlit, GenericOffset, and SpectroAstrometry templates.

- `SEQ.CUMOFFSETX` and `SEQ.CUMOFFSETY` for cumulative offsets in pixels.
- `SEQ.CUMOFFSETA` and `SEQ.CUMOFFSETD` for cumulative offsets in arcseconds (alpha, delta).

Cumulative offsets are always relative to the **first** frame in the batch (`TPL.EXPNO=1`).

If the same guide star is used before and after an offset, the offsetting accuracy is about 0.1 arc seconds. All recipes looking for offset information take this into account and will use the header offset information as a first guess and will refine the offset through cross-correlation techniques.

The jitter offsets are generated using a Poisson distribution. `SEQ.POISSON` refers to the number of Poisson numbers which have been generated at the start of the jitter sequence. See the eclipse web page (<http://www.eso.org/projects/dfs/papers/jitter99/>) for more information about this factor.

2.8 Pipeline Products

To allow identification of pipeline products, some keywords are inserted in the output FITS headers.

- `PIPEFILE` is a standard 8-char FITS keyword which contains the name of the file as set by the pipeline when creating the product, it is useful as a label to identify the file.
- `PR0.DID` contains the version number of the dictionary used for all product keywords.

¹Templates might have been aborted, so it is possible to have fewer frames than `TPL.NEXP` in a batch.

- `PRO.SCIENCE` contains a boolean to identify 'SCIENCE' products.
- `PRO.TYPE` contains the type of the data products as one of 'COMBINED', 'DARK', 'SPEC_PIX', 'SPEC_WL', etc. replaces
- `PRO.TYPE` contains the type of the data products as one of 'TEMPORARY', 'PREPROCESSED', 'REDUCED', or 'QCPARAM'.
- `PRO.DATE` gives the date and time of the pipeline execution for the frame creation.
- `PRO.REC1.ID` identifies the recipe that generated the file, with a unique name.
- `PRO.REC1.DRS.ID` identifies the Data Reduction System that was used to produce the file (CRIRES pipeline for all CRIRES recipes).
- `PRO.DATANCOM` specifies the number of raw frames that were combined to generate the product. Its exact meaning depends on the recipe, see each recipe documentation to learn what it refers to.

2.9 Summary Table

The following table summarises all keywords introduced in this chapter, in alphabetical order.

FITS keyword	Meaning
<code>ARCFILE</code>	Archive file name
<code>DEC</code>	Declination (J2000) in degrees
<code>DET.DIT</code>	Detector Integration Time
<code>DET.NDIT</code>	Number of (averaged) DITs
<code>DPR.CATG</code>	Data product category
<code>DPR.TYPE</code>	Data product type
<code>DPR.TECH</code>	Data product acquisition technique
<code>INS.DROT.MODE</code>	Derotator mode (e.g. SKY or ELEV)
<code>INS1.DROT.POSANG</code>	Position angle on sky (degrees)
<code>INS.GRAT.ORDER</code>	Grating order
<code>INS.LAMP1.ST</code>	Status of the Halogen lamp
<code>INS.LAMP2.ST</code>	Status of the IR emitter lamp
<code>INS.LAMP3.ST</code>	Status of the NeKr lamp
<code>INS.LAMP4.ST</code>	Status of the Thorium–Argon lamp
<code>INS.SLIT1.WID</code>	Entrance slit width (since Oct 2011 set to 0.2 or 0.4 arcseconds).
<code>INS.WLEN.REF</code>	Reference wavelength (microns)
<code>INS.WLEN.CWLEN</code>	Central wavelength (microns)
<code>INS.WLEN.END1</code>	Wavelength at the end of detector 1 (microns)
<code>INS.WLEN.END2</code>	Wavelength at the end of detector 2 (microns)
<code>INS.WLEN.END3</code>	Wavelength at the end of detector 3 (microns)
<code>INS.WLEN.END4</code>	Wavelength at the end of detector 4 (microns)
<code>INS.WLEN.MIN</code>	Minimum wavelength of the unvignetted spectral range (microns)
<code>INS.WLEN.MAX</code>	Maximum wavelength of the unvignetted spectral range (microns)
<code>OBS.PROG.ID</code>	Program ID
<code>OBS.NAME</code>	Observation Block name

OBS.TARG.NAME	Target package name
ORIGFILE	File name given during acquisition
PIPEFILE	File name as attributed by the pipeline
PRO.CATG	Data product category
PRO.DATANCOM	Number of frames combined into this product
PRO.DATE	Pipeline reduction date
PRO.DID	Dictionary version for PRO keywords
PRO.RECi.ID	Applied recipe ID
PRO.RECi.DRS.ID	Data Reduction System ID
PRO.REDLEVEL	Data product reduction level
PRO.STATUS	Data product reduction status
PRO.TYPE	Data product type
RA	Right ascension (J2000) in degrees
SEQ.CUMOFFSETX	Cumulative offset in X, in pixels
SEQ.CUMOFFSETY	Copulated offset in Y, in pixels
SEQ.CUMOFFSETA	Emulated offset in alpha, in arcseconds
SEQ.CUMOFFSETD	Accumulated offset in delta, in arcseconds
SEQ.POISSON	Poisson homogeneity in jitter throw
TEL.AIRM.START	Airmass at start of exposure
TEL.AIRM.END	Airmass at end of exposure
TEL.AMBI.FWHM.START	ASM seeing at start
TEL.AMBI.FWHM.END	ASM seeing at end
TPL.ID	Template ID
TPL.NEXP	Number of exposures expected in the template
TPL.EXPNO	Exposure rank in the template

DIT (s)	NDSAMPLES
1 to 9	2
10, 15	4
20	6
30	8
45	10
60	14
90	20
120	28
180	40
300	66
600,900	128

Table 1: NDSAMPLES per DIT for CRIRES.

3 CRIRES data and calibrations

CRIRES uses a science detector which is a 4×1 mosaic of four Raytheon 1024x1024 pixel InSb Aladdin III detectors. When reducing data you should keep in mind the following points.

- The arrays are read out simultaneously.
- Only the bottom half of each detector is illuminated by the slit, so the useful mosaic actually consists of $4 \times 1024 \times 512$ pixels. The reading-out is executed parallel to the mosaic for detector #2 and #3, and perpendicular to it for detectors #1 and #4.
- There is no shutter, so the array is continuously exposed to light.

3.1 Readout modes

The CRIRES IRACE controller and the associated DCS software control the readout of the array. Currently, two read-out modes are supported: *Fowler Sampling* (FNS) and *Fowler Sampling with Global Reset and Window* (FowlerNsampGRstWin). In FNS mode, the detector is first reset, then each pixel is non-destructively read NDSAMPLE/2 times at the start of the exposure, then again NDSAMPLE/2 at the end of the exposure time DIT. The slope of the relation provides the intensity of the given pixel. NDSAMPLE varies with the DIT: 2 for $\text{DIT} < 10$ sec, 4 for $10 \text{ sec} \leq \text{DIT} \leq 20$ sec etc... up to 128. In case of very bright targets, the *FowlerNsampGRstWin* mode can be used. This mode only reads detector #2 and #3. It executes a global reset of the detector before applying the Fowler Sampling on two small stripes that fully cover detector #2 and #3 in the spectral direction but only a user selected number of pixels (64, 128 or 256) in the spatial direction. As only a small fraction of the detector is read, shorter DITs can be used. On the other hand, for comparable DITs, this mode allows to reduce the read-out noise by a factor ~ 2 compared to the FNS mode, hence observations of faint targets are less affected by the read-out noise. Table 1 shows the NDSAMPLES value for each DIT. The information is also present on the following webpage:

http://www.eso.org/sci/facilities/paranal/instruments/crides/doc/DIT_NDSAMPLE.html

Fig. 2 shows an example of windowed mode *FowlerNsampGrStWin* (only strips of Detectors 2 and 3 are read out) with Fig. 6 showing an example of "normal" readout mode *Fowler Sampling*.

3.2 EXPTIME, NDIT and DIT

Up until March, 2000, the ESO EXPTIME keyword was defined as $\text{NDIT} \times \text{DIT}$. This caused some confusion as the frames are the average and not the sum of NDIT exposures. After March, 2000, the EXPTIME keyword is set to the DIT.

3.3 Calibrations

3.3.1 Darks

Figure 3 shows dark images taken with a 1-s and 600-s exposure time. Long darks show **detector glow** as is very obvious from this figure. The calibration plan specifies 3 darks taken per DIT setting. Visitors should be reminded to avoid very long DITs of slightly different value (e.g. 600-s, 650-s, 700-s), due to the extra calibration load this imposes. Note that any DIT is comprised of a number of NDSAMPLES as shown in the [CRIRES webpages](#). These values are different depending on the readout mode used and are shown in Table 1.

Note that although some DITs have much smaller readout noise in the windowed mode *FowlerNsampGrStWin*, at very low or very high DITs the RMSs are similar. See the User Manual for details.

3.3.2 Flatfields

For flatfields the Halogen lamp or infrared emitter is used. The S/N ratio of the flatfields can be found [here](#). There may be large gradients from detector 1 to detector 4 due to the position of the centre of the order and/or the presence of absorption lines caused by water vapour in the path between the lamp and the entrance window and/or vignetting by the intermediate slit.

A flatfield with reference wavelength of 5050.4-nm is shown in Fig. 4. Note that many flats, such as the one displayed, show **absorption due to water vapour in the instrument** between the calibration lamp and the vessel entrance window. Flatfields are taken with a 0.2 arcsec slit. Some flats show **ghosts**, for example visible in Fig. 5.

Until now flatfields have been taken daily by the observatory for each wavelength setting used although it is possible that this calibration frequency may decrease in the future.

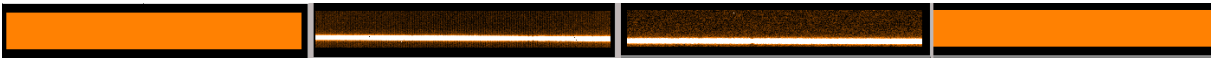


Figure 2: Example of windowed mode. Note that only detectors 2 and 3 are read out and there is no flux on detectors 1 or 4.

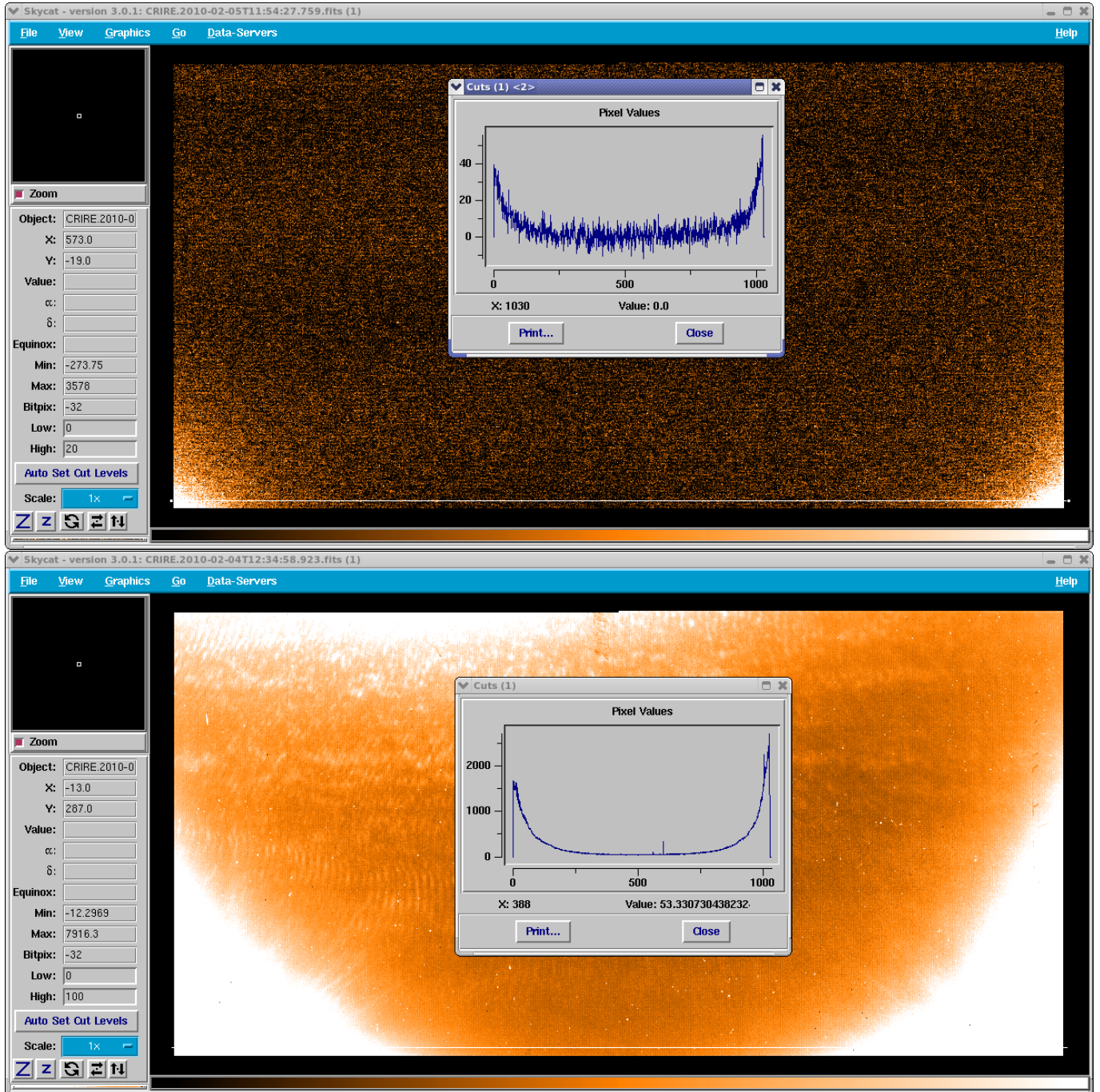


Figure 3: Top image: A 1-s dark (detector 3, range 0 to 20 ADU). Bottom image: A 600-s dark (detector 3, range 0 to 100 ADU). Note the strong detector glow present and is visible on all four detectors. The cut is taken at $y=10$ pixels.

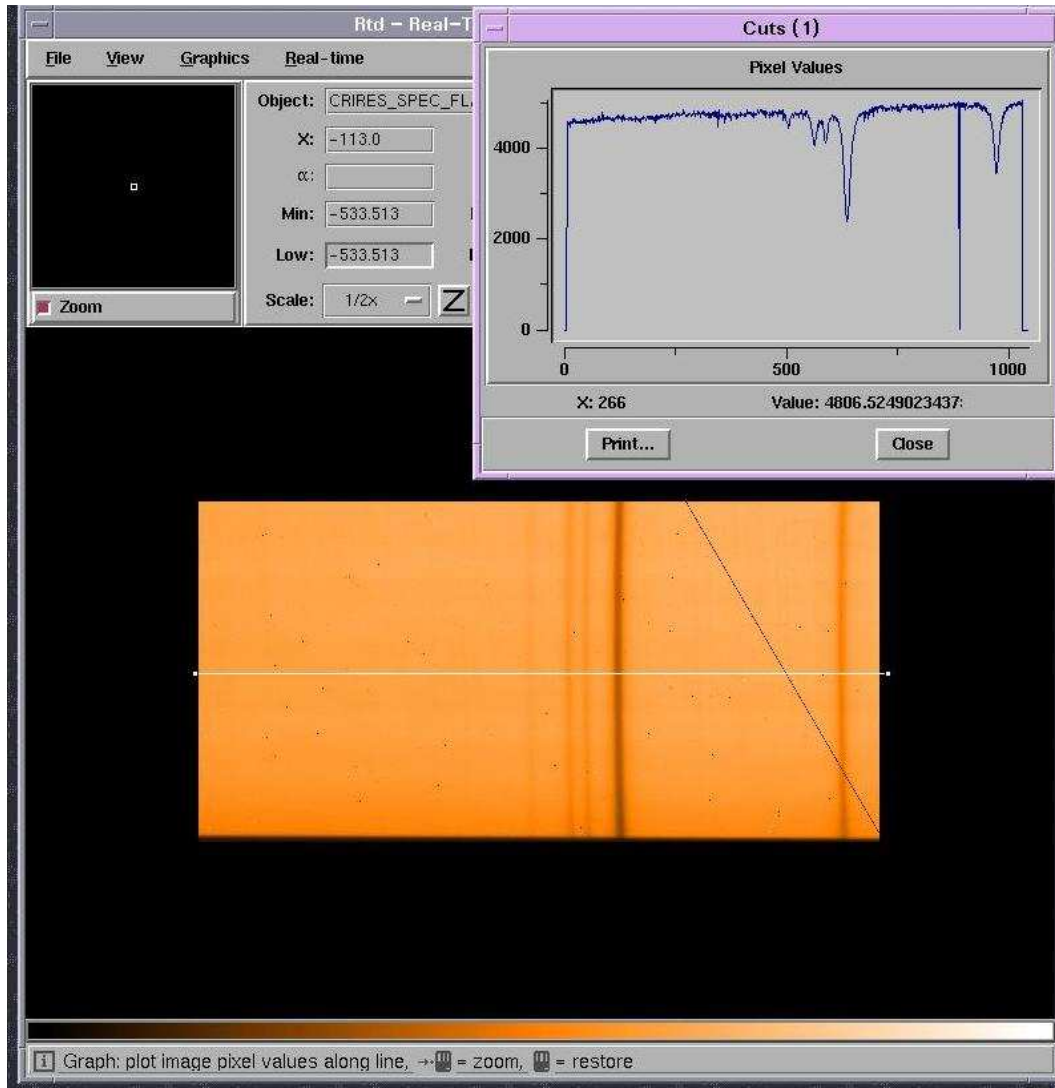


Figure 4: Science detector image of a flatfield on detector 3 at $\lambda=5050.4$ -nm. The diagonal black line is caused by bad pixels. The absorption lines in the spectrum are due to water vapour in the instrument between the calibration lamp and the vessel entrance window.

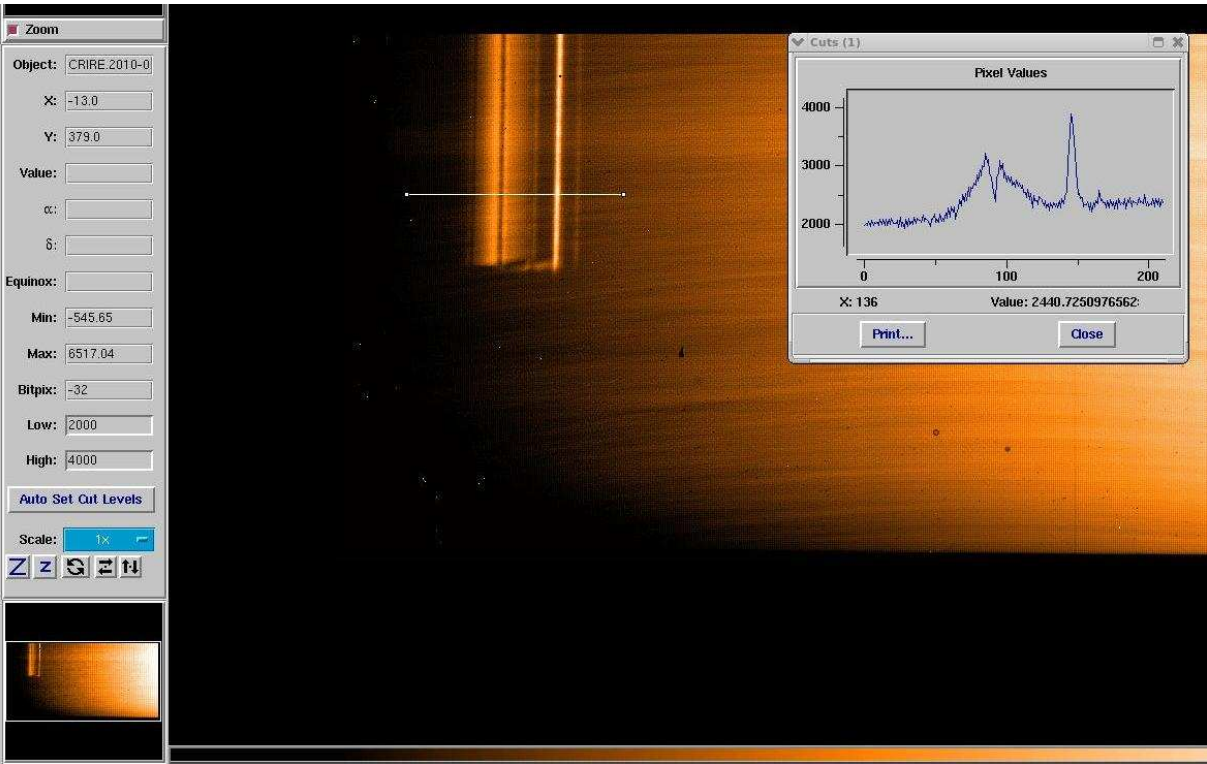


Figure 5: Science detector image of a flatfield at 1561-nm that shows strong ghosts.

3.3.3 ThAr wavelength calibration

ThAr wavelength calibration is typically performed using the `crires_spec_cal_wave` template and using a 0.2 arcsec slit. Because of problems with scattered light, this calibration is now performed using fibres. Note that ThAr calibration is only performed for wavelengths **shorter than 2400-nm**. The DIT and NDIT are calculated automatically from table `cmm-cfgMaxFluxThAr.dat` (which contains ThAr intensities every twenty pixels). This aims at avoiding remnance caused by very bright ThAr lines.

An example of a ThAr calibration on the science detector is shown in Fig. 6. Note that as well as the six ThAr fibres there are also **metrology fibres** at the top and bottom of the science detector. These fibres do not go through the same optical path as the six ThAr fibres, hence are not aligned in wavelength with the main fibres. The fibres at the "top" pass through the grating only, those at the bottom through the prism and the grating.

ThAr calibrations are taken daily by the observatory for each wavelength setting used the previous night (for $\lambda < 2400\text{-nm}$). In future the frequency of these calibrations may be reduced. Any such change will be announced in the [CRIRES news pages](#).

Note that due to bad pixels and the scratch on detector 2 and other detector features, false lines may be detected in the wavelength solution that cause a decrease in the correlation. Such features can be removed by masking before doing the wavelength calibration. We note at present this is not possible within the CRIRES pipeline although may be rectified in a future release.

3.3.4 Gas-cell wavelength calibrations

Wavelength calibration can also be performed by taking a flatfield but with the gas-cell inserted. This produces an absorption-line spectrum in the flatfield as shown in Fig. 7. An N₂O line list is provided that can be used for wavelength calibration using `cires_spec_wavecal`. Gas-cell calibrations are taken daily by the observatory for each wavelength setting used the previous night.

3.3.5 Detector linearity images

Typically every 1–2 months, flatfield calibrations are taken with the IR-emitter at 4007.0-nm at a variety of DIT settings (1,2,3,4,5,6,7,8,9,10,20,30... 80,90,100-s) in order to determine the non-linearity coefficients of the detector (so called "Detector monitoring"). Fig. 8 shows an example of such flatfields taken with 2.0-s and 100.0-s DITs.

These non-linearity images are processed in Quality Control, Garching, and are available here: <http://www.eso.org/observing/dfo/quality/CRIRES/pipeline/pipe.calib.html>.

If you are looking at pre-April-2011 CRIRES data then you need ONE file with data category `COEFFS_CUBE`. If you are looking at post-April-2011 CRIRES data then you need THREE files, `DETLIN_A`, `DETLIN_B`, and `DETLIN_C`. These files should be included within your reduction blocks to correct for non-linearity.

Note that there is the complication that Since version 2.1.0 of the pipeline (released in April 2011), the DETLIN correction coefficients are computed by a different recipe (package `detmonp` - recipe `detlin`) and the coefficients format has changed. The pipeline expects all coefficients to be stored in the same FITS cube with TAG (or `PR.CATG`) `COEFFS_CUBE`

Before 2.3.0, the pipeline expected 3 files (1 file per coefficient) with the TAGs (or `PRO.CATG`)

DETLIN_A, DETLIN_B, DETLIN_C.

If you are using 2.3.1, you must provide the detlin correction coefficients in the new format (the cube containing PRO.CATG=COEFFS.CUBE).

If you have old format coefficients files that you want to use, you need to regenerate the coefficients using the `detmonp detlin` recipe on the old detector linearity raw calibration data. There used to be a `crires.util.convert` recipe in the pipeline to convert old coefficients files into the new ones. But it has been removed. The `detmon` package is available here <http://www.eso.org/sci/software/pipelines/detmon/detmon-pipe-recipes.html>

3.4 General Features of CRIRES Infrared spectra

3.4.1 Glow

The 4 detectors show glow over areas close to the amplifiers, as clearly visible in Fig. 3. It is stable so can be subtracted by using exposures taken with the same DIT. Therefore, in order to correct for this detector defect, it is recommended that observations at any wavelength setting should include at least one nodding cycle.

3.4.2 Optical ghosts

Occasionally, especially when using the CRIRES Echelle grating in high order, a quasi white-light ghost is visible on the data. It results from a retro-reflection from the detector onto the grating, which is redirected onto the detector in a different order and with reduced dispersion. Ghosts are canceled out by nodding, as sky emission lines do. An example of ghost is shown in Fig. 9.

3.4.3 Odd-even column effect and detector non-linearity

Images obtained with the science detector show a pattern aligned with the reading direction: parallel to the rows of detector #2 and #3 and to the columns of detector #1 and #4. This pattern, called *odd-even effect* consists of alternating rows or columns of larger and smaller intensities than the mean value and it depends on the signal level. This effect cannot be removed by simply flat-fielding and it is still present after nodding. However, if during your observations you managed to keep the total signal (source + sky + dark) below 10,000 ADUs ($\approx 80,000e^-$) and DIT of both science and flat field, larger than 2 sec, then you can still get a good correction (with residuals smaller than $\sim 1\%$) of the non-linearity of the detectors, including the odd-even effect, by downloading non-linearity corrections appropriate to the data of observation from the following webpage:

http://www.eso.org/observing/dfo/quality/CRIRES/pipeline/pipe_calib.html Data taken with DITs shorter than 2 sec are affected by the finite reset time of the detector arrays. Sections 8.3 and 10.4 describe how to obtain the non-linearity coefficients and their limitations at low- and high- light levels.

3.4.4 Remnants from bright sources

If the array is exposed to a bright source and is saturated, a remnant becomes visible. The amplitude of the remnant and the time it lasts depend on the brightness of the source; see

Fig. 10 for an example. They are a nuisance, and, by Murphy's law, are bound to land on your spectrum. Hence in Service Mode no saturation of the detectors is permitted.

3.4.5 Fringing

No extreme detector fringes are detected in the wavelength range useful for science observations. Strong fringes are only visible beyond 5 μm , outside the atmospheric window.

4 Downloading CRIRES raw data and calibrations

Users must have an ESO user portal ID, that is also used to submit proposals. It is easily obtained by registering at <http://www.eso.org/UserPortal/>. Once registered, raw CRIRES spectra and calibrations should be downloaded from the ESO archive at:

<http://archive.eso.org/wdb/wdb/eso/crides/form>. Figures 11 to 13 show the different steps in the download process.

- Figure 11 shows the initial search. One can search one or several parameters, including (but not limited to) programme ID, target name, night (or range of nights), coordinates in RA/Dec or Galactic systems (with or without search box), and/or proposal title. A list of targets can also be used as input. By default no sorting is performed on the output although it may be convenient to sort by observation time or type. One can also increase the number of rows returned from the default value of 200.
- Figure 12 shows the result of the search. If the check boxes on the left are green then the spectra can be downloaded, if red then they are still within their proprietary period (typically 1 year) and can only be accessed by the P.I. (or in the future possibly CoIs given permission by the P.I.). Mark the spectra that you would like to download.
- Figure 13 shows the result of the request. At this point one should normally click the "Cal selector" and "Add IRAF keywords" boxes. The former searches the ESO archive for calibrations close in time to the science frames. The latter adds IRAF keywords to the headers.
- Figures 14 and 15 show the results of the request but now with the associated calibrations.
- Finally, figure 16 shows the java Applet that is used to retrieve the science frames and associated calibrations and put them on your computer. For Unix/Linux users a shell script is also created that can be downloaded and run in order to easily fetch all of the data.
- Once the download is complete, you are ready to inspect the science and calibration data and reduce your data.

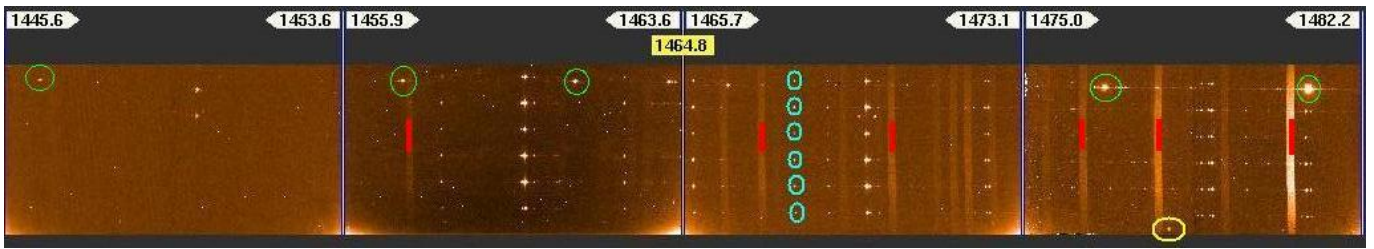


Figure 6: Science detector images whilst a ThAr image is being taken. Light blue-green circles: examples of ThAr lines for each of the six fibres. Green circles: Metrology fibres. Yellow circles: Metrology fibres. Note that the metrology fibres do not go through the same optical path as the six ThAr fibres, hence are not aligned in wavelength. Vertical red lines: stray light generated by strong spectral features falling close to the intermediate slit border.

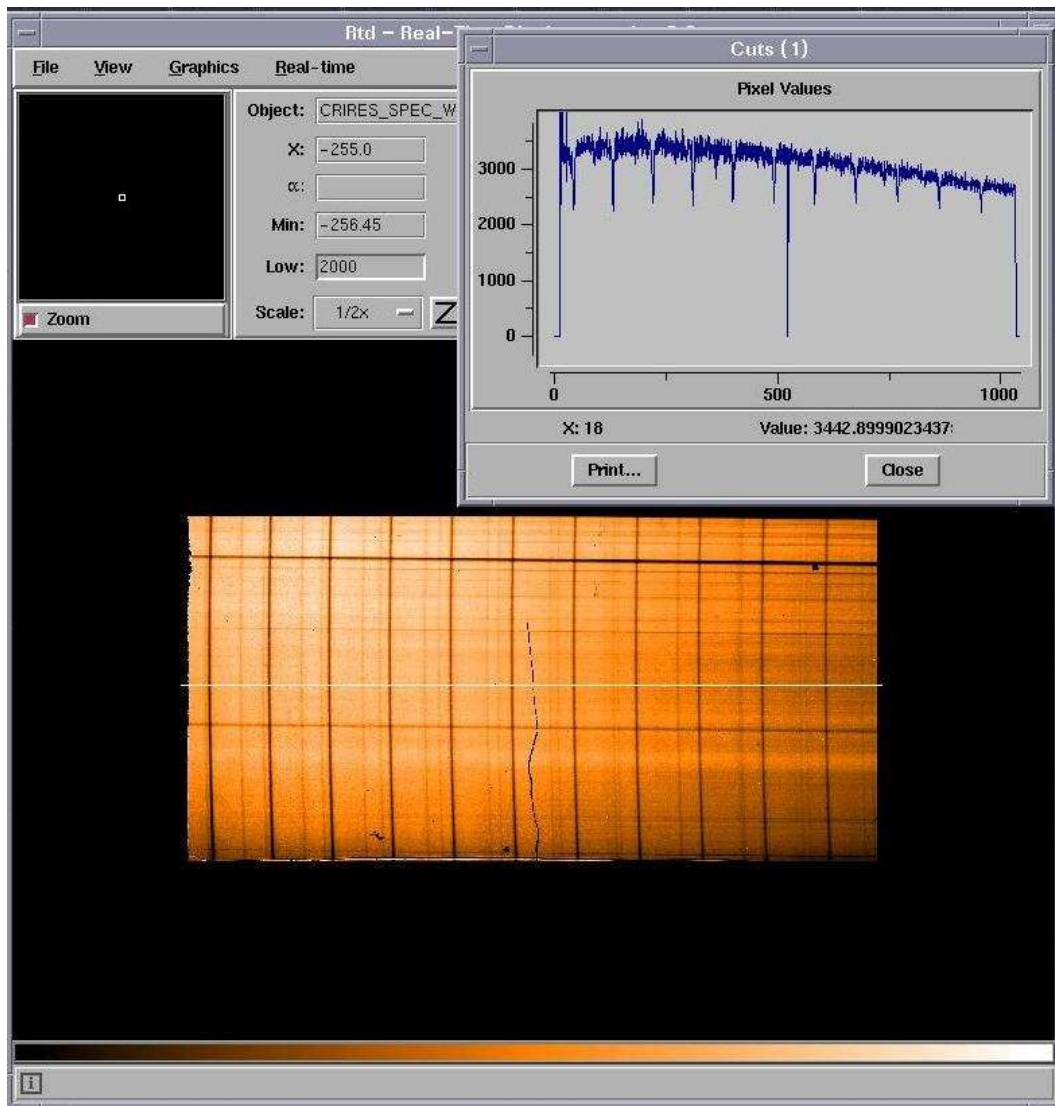


Figure 7: Science detector image on detector 4 of a flatfield with the N20 gas cell inserted. This produces a sharp-lined absorption-line spectrum that can be used for wavelength calibration.

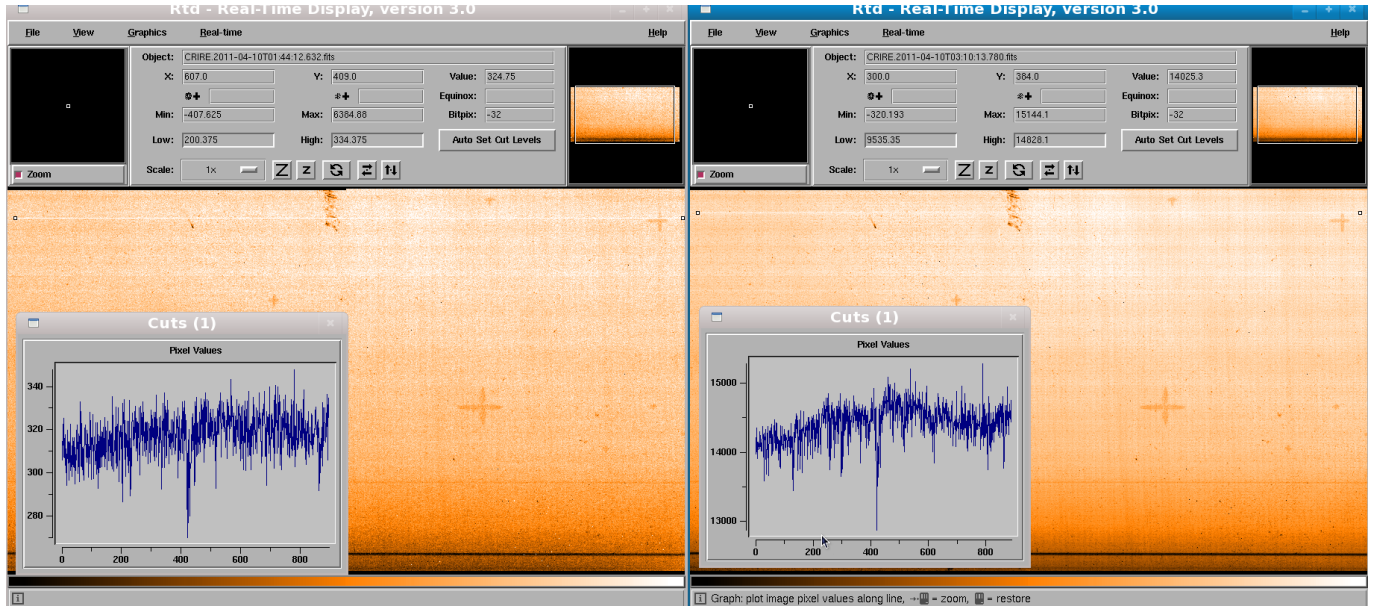


Figure 8: Detector linearity frames taken at 4007.0-nm. Left panel: DIT=2.0-s. Right panel: DIT=100.0-s.

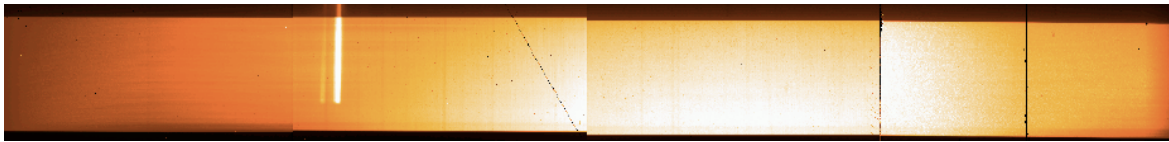


Figure 9: Example of ghost observed with a reference wavelength of 1153.8 nm. The ghost is visible in a staring frame top left on detector #2. Also a diagonal scratch (right edge of detector #2) and a vertical scratch (middle of detector #4) are visible.

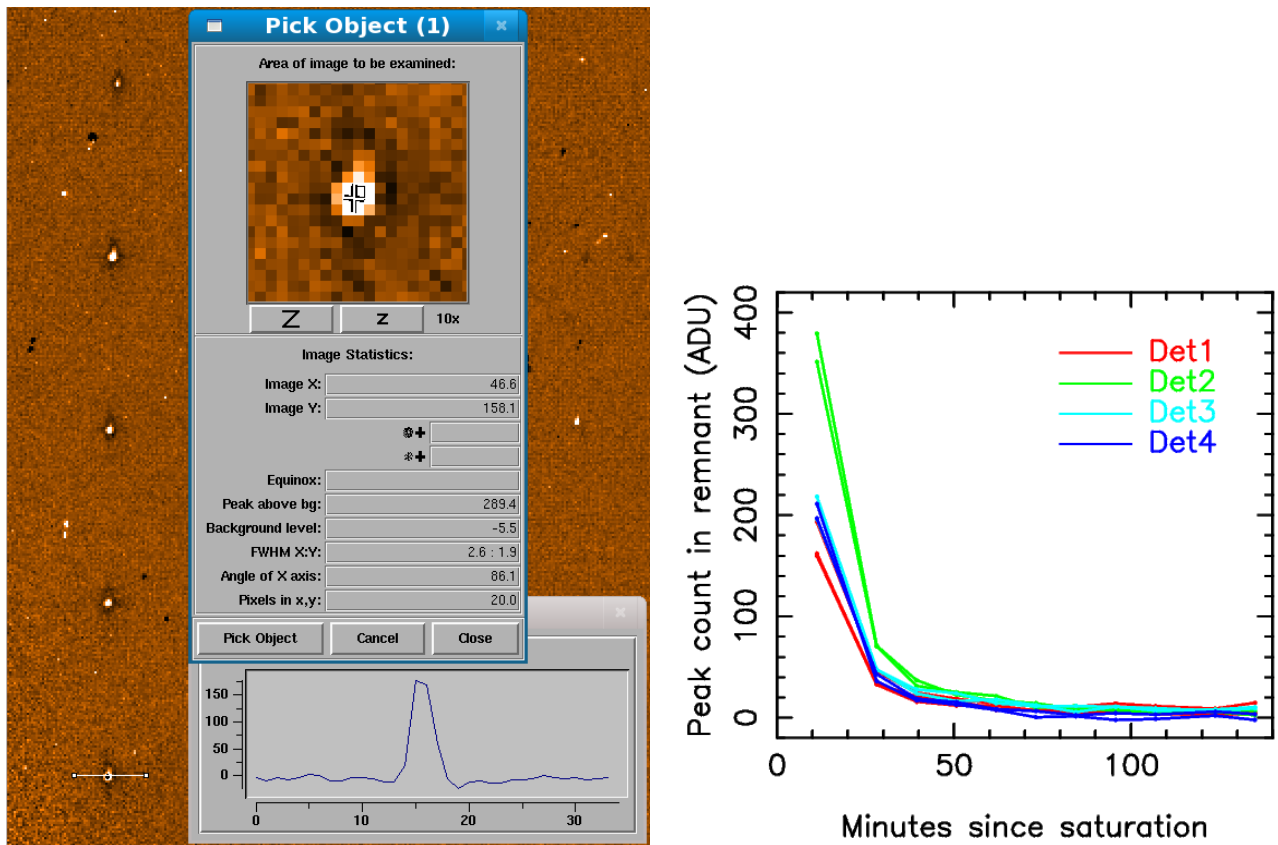


Figure 10: Left plot: Example of remanence on a 600-s dark image on Detector 1 taken after the detector was saturated with a ThAr arc exposure. Right plot: Decay of remanence with time for the four detectors.

ESO Science Archive - CRIRES Data Products - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://archive.eso.org/wdb/wdb/eso/crises/form

ADS APOD NEWS AstroPH Goog SIM ESO FLA FLA INT FLA QC JVS_Sch PSO PSO Int Remedy CRIRES CRIRES Int

Resorting Query ... ESO - Paranal Ins... ESO - PSO home ESO Science Arc... CRIRES - SCIOPS ...

ESO Science Archive CRIRES Query Form

Other Instrument Query Forms

Archiva Facility HOME ESO HOME Form INFO FAQ

Description

This form provides access to observations performed with CRIRES that are stored on-line in the Science Archive Facility in Garching. Find more information about CRIRES data on the [Quality Control web pages](#).
Paranal ambient conditions can be queried separately from [here](#).

Search Reset query Help Status of Requests

Target Name and Equatorial Coordinates

Target.....: HD 49331 SIMBAD name

☐ Input Target List... Browse...

Search Box.....: 00 10 00 If Simbad/Net name or coordinates given

☒ RA.....: DEC..... (J2000) Format RA(h) DEC(deg)

Galactic Coordinates

☐ Galactic Search Box : 00 30 00 If galactic coord. given

☐ Galactic Longitude : Galactic Latitude degrees

Observation and proposal parameters

☐ Night.....: (DD MM YYYY of night begin (12:00 UT))

OR give a query range using the following two fields (start/end dates)

Start.....: 01 01 2011 12 hrs [UT] End 01 06 2011 12 hrs [UT]

☒ ProgId.....: PPP.C-NNNN (e.g. 073.C-0156*)

☐ Program Type.....: Any SV Mode All modes

☐ PI/CoI.....: as PI only

☐ Proposal Title.....:

Generic File Information

☒ DP_ID.....: archive filename of FITS file (e.g. CRIRES.2010-03-13T00:00:500)

Figure 11: Initial steps in downloading data from the ESO archive, searching for CRIRES observations of HD 49331 between 1st Jan 2011 and 1st June 2011.

ESO Science Archive - CRIRES Data Products - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://archive.eso.org/wdb/wdb/eso/crises/query

ADS APOD NEWS AstroPH Goog SIM Y ESO FLA FLA INT FLA QC JVS_Sch PSO PSO Int Remedy CRIRES CRIRES Int

Resorting Query ... ESO - Paranal Ins... ESO - PSO home ESO Science Arc... CRIRES - SCIOPS ...

ESO Science Archive CRIRES Query Results

Other Instrument Query Forms

Archive Facility HOME ESO HOME Form IMEC FAD

To request datasets please mark the datasets, then press either the **Instant download** button or the **FTP or media** delivery button.
(You will be prompted for your ESO User Portal username and password. If you do not yet have an ESO User Portal account, please fill out the [registration form](#).)
Datasets for which the proprietary period is over are highlighted in **green** and are publicly available.
Datasets that are still under the proprietary period are highlighted in **red** and can only be downloaded by the corresponding PI.
Any comments or suggestions for improvement of this form can be sent to archive@eso.org.

Define new query Status of Requests

SIMBAD coordinates for HD 49331 : 06 47 37.2, -08 59 54.6

MarkAll UnMarkAll MarkAllPublic

Instant download FTP or media Reset

M	HDR	Release Date	Object	RA	DEC	ProgId	DP.ID	OB ID	DPR CATG	DPR TYPE	DPR TECH	OBS Target name	INS_MODE
☒	Header	2011-01-15	HD-49331	06:47:36.94	-08:59:52.5	086.D-0066(A)	CRIRE.2011-01-15T04:46:56.780	500553	SCIENCE	OBJECT	SPECTRUM,NODDING,OTHER	HD-49331	Spectroscopy
☒	Header	2011-01-15	HD-49331	06:47:37.57	-09:00:04.3	086.D-0066(A)	CRIRE.2011-01-15T04:47:50.972	500553	SCIENCE	OBJECT	SPECTRUM,NODDING,OTHER	HD-49331	Spectroscopy
☒	Header	2011-01-15	HD-49331	06:47:36.94	-08:59:52.7	086.D-0066(A)	CRIRE.2011-01-15T04:49:29.646	500553	SCIENCE	OBJECT	SPECTRUM,NODDING,OTHER	HD-49331	Spectroscopy
☒	Header	2011-01-15	HD-49331	06:47:37.59	-09:00:04.2	086.D-0066(A)	CRIRE.2011-01-15T04:50:27.917	500553	SCIENCE	OBJECT	SPECTRUM,NODDING,OTHER	HD-49331	Spectroscopy
☒	Header	2011-01-15	HD-49331	06:47:36.93	-08:59:52.8	086.D-0066(A)	CRIRE.2011-01-15T04:52:00.754	500553	SCIENCE	OBJECT	SPECTRUM,NODDING,OTHER	HD-49331	Spectroscopy

Figure 12: Result of ESO archive search shown in Fig. 11. Two nodding spectra have been selected.

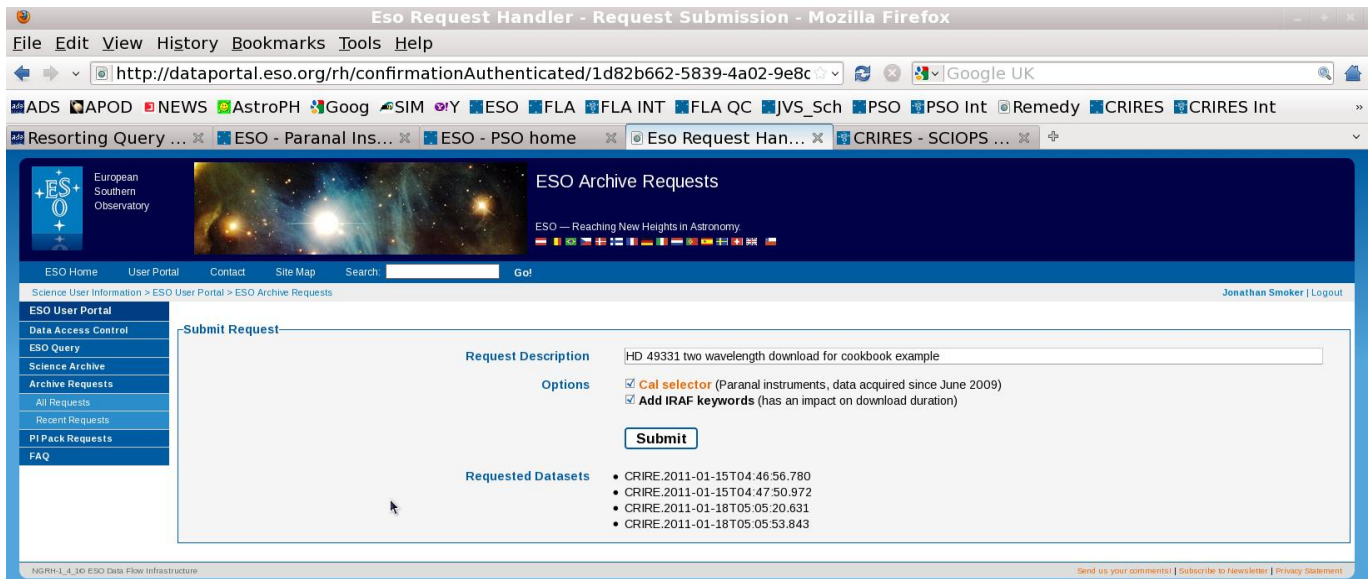


Figure 13: Intermediate steps in downloading data from the ESO archive. Clicking the "cal-selector" tab will query the archive for calibrations needed to reduce the science data.

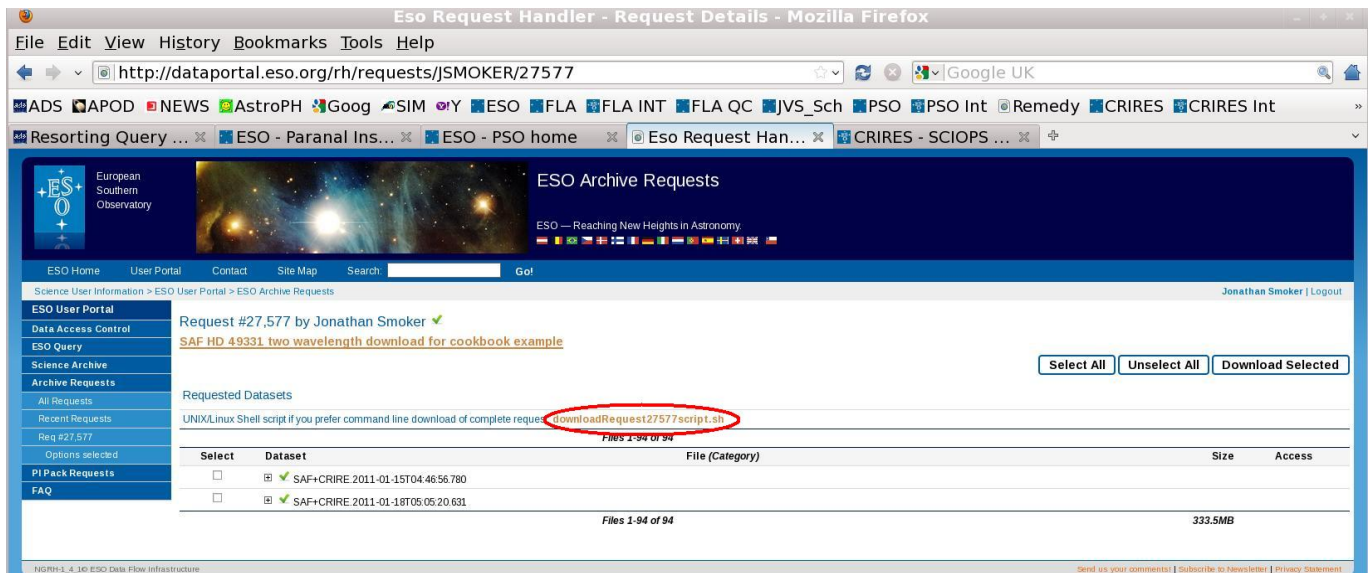


Figure 14: Query result, hitting "+" will show the associated calibrations that were associated with the science spectra queried in Fig. 13. A shell script can be saved to download the data and calibrations (red ellipse) or a java applet run (Fig. 16).

ESO Request Handler - Request Details - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://dataportal.eso.org/rh/requests/JSMOKER/27577

ADS APOD NEWS AstroPH Goog SIM Y ESO FLA FLA INT FLA QC JVS_Sch PSO PSO Int Remedy CRIRES CRIRES Int

Resorting Query ... ESO - Paranal Ins... ESO - PSO home ESO Request Han... CRIRES - SCIOPS ...

ESO Home User Portal Contact Site Map Search: Gof

Science User Information > ESO User Portal > ESO Archive Requests Jonathan Smoker | Logout

Request #27,577 by Jonathan Smoker ✓

SAF HD 49331 two wavelength download for cookbook example

Select All Unselect All Download Selected

Requested Datasets

UNIX/Linux Shell script if you prefer command line download of complete request: [downloadRequest27577script.sh](#)

Select	Dataset	File (Category)	Size	Access
☐	SAF+CRIRE 2011-01-15T04:46:56.780	CRIRE.2011-01-15T04:46:56.780.fits.Z (OBS_NOD)	4.0MB	✓
☐		CRIRE.2011-01-15T04:47:50.972.fits.Z (OBS_NOD)	4.0MB	✓
☐		CRIRE.2011-01-15T04:45:54.946.fits.Z (ACQUISITION)	1.4MB	✓
☐		CRIRE.2011-01-15T10:03:52.997.fits.Z (CAL_FLAT)	5.6MB	✓
☐		CRIRE.2011-01-15T12:33:25.959.fits.Z (CAL_FLAT)	5.9MB	✓
☐		CRIRE.2011-01-15T12:34:45.559.fits.Z (CAL_FLAT)	5.9MB	✓
☐		CRIRE.2011-01-15T12:36:09.493.fits.Z (CAL_FLAT)	5.9MB	✓
☐		CRIRE.2011-01-15T13:48:41.291.fits.Z (CAL_DARK)	6.1MB	✓
☐		CRIRE.2011-01-15T14:05:22.521.fits.Z (CAL_DARK)	6.0MB	✓
☐		CRIRE.2011-01-15T22:59:15.665.fits.Z (CAL_DARK)	4.5MB	✓
☐		CRIRE.2011-01-15T23:04:21.902.fits.Z (CAL_DARK)	4.5MB	✓
☐		M.CRIRES.2008-06-17T16:47:59.528.fits (CALPRO_HITRAN_CATALOG)	8.0MB	✓
☐		M.CRIRES.2008-06-17T16:48:16.005.fits (CALPRO_OH_CATALOG)	272.8KB	✓
☐		M.CRIRES.2008-06-17T16:48:31.605.fits (CALPRO_MODEL_CONFIG)	11.2KB	✓
☐		M.CRIRES.2010-09-21T17:04:57.320.fits (DETUN_A)	8.1MB	✓
☐		M.CRIRES.2010-09-21T17:05:07.420.fits (DETUN_B)	8.1MB	✓
☐		M.CRIRES.2010-09-21T17:05:13.214.fits (DETUN_C)	8.1MB	✓
☐		CRIRE.2011-01-15T04:46:56.780.NL.txt (NIGHTLOG_INFO)	666B	✓
☐		CRIRE.2011-01-15T04:46:56.780.xml (ASSOCIATION_TREE)	8.9KB	✓
☐	SAF+CRIRE 2011-01-18T05:05:20.631			

Files 1-94 of 94

333.5MB

NRH-3_1_36 ESO Data Flow Infrastructure

Send us your comments | Subscribe to Newsletter | Privacy Statement

Figure 15: Results of query of the raw data and associated calibrations that can either be downloaded using the java applet or the shell-script highlighted by the red ellipse shown in Fig. 14.

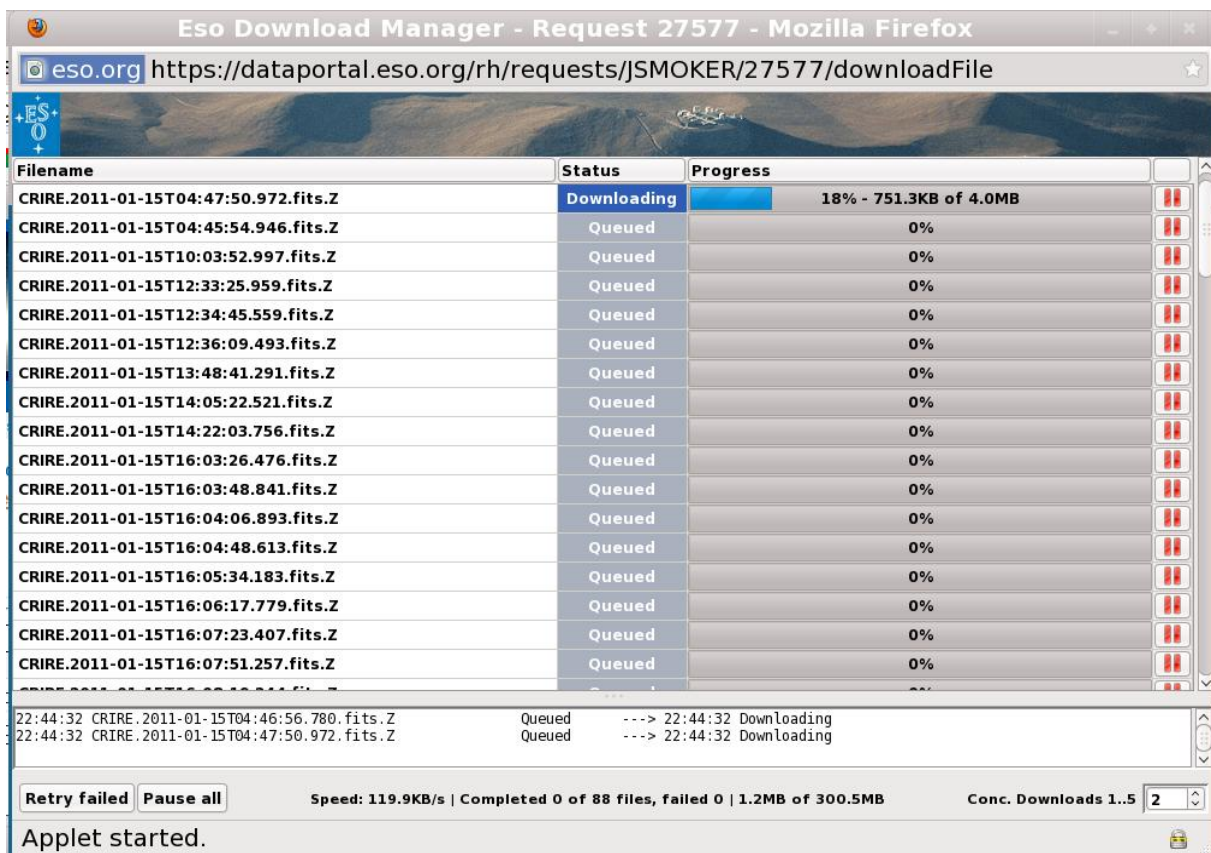


Figure 16: Final download to the users computer using the java applet. The shell script highlighted in Fig. 15 can also be used.

5 Reducing the science data

The most basic way of taking IR spectra is to observe the target along two slit positions (i.e. nodding along the slit). The sky is then removed by a process which is sometimes called *double sky subtraction*. The basic steps of how to reduce these type of data are:

1. First subtraction, Ghost and Glow Removal
2. Flat fielding
3. Wavelength calibration
4. Combining 2d spectra
5. Extraction
6. Telluric line correction
7. Flux calibration - by default flux standards are not part of the CRIRES calibration plan at each setting taken, only five settings are frequently observed with reference wavelengths 1087.3, 1504.8, 2205.0, 3611.0 and 5050.4-nm. The results are available here: [here](#).

5.1 First subtraction

There are two techniques observers use in taking IR spectra: *i)* the classic sequence where one observes the object at two slit positions (i.e. ABBA, *AutoNodOnSlit* template); and *ii)* the more complex case where one has observed the target along several slit positions (i.e. *GenericOffset* template).

In case *i)*, one simply subtracts frames taken at different slit positions. So one needs to form A-B, B-A couples and then combine them together. This simple step removes the dark and results in an image with 3 spectra, one positive and 2 negative. Moreover, as mentioned in section 3.4, also the glow and ghost defects are canceled out.

In case *ii)*, one could build a sky frame from several spectra, as one does when building the sky frame in imaging. This results in an image with only one positive spectrum. In this case you will might not be able to correct for ghost and glow since the obtained sky would result from the median of your spectra, so it will not contain the same glow and ghost features as your science frames.

5.2 Flat fielding

For each wavelength setting three lamp flat images are taken. The CRIRES pipeline recipe `crires_spec_flat` provides a *master flat* obtained as follows:

- normalise each single flat to the median value of its central part not affected by vignetting. In case the vignetting region is actually different from the one expected, one can change the value of the *starty* and *stopy* parameter to opportunely set the zone where to compute the median used to normalise the single flat lamps.

- divide each frame by a 2nd-degree polynomial whose coefficients are part of the static calibration files specifically obtained to correct for non-linearity effects, and available at http://www.eso.org/observing/dfo/quality/CRIRES/pipeline/pipe_calib.html
- average the resulting three flats

The recipe does not remove the wavelength dependence of the lamp from the flat field. This is removed when the telluric standard is divided into the spectrum of the science target. Implicit in this statement is that the same flat was used for both the target and the standard and that the target and the standard were observed at roughly the same place on the array. If neither of these things are true then flux calibration (relative and absolute) will be less accurate.

The described procedure can be also done by using some IRAF's tasks, such as *flatcombine*, under the *mscred* environment.

5.3 Wavelength calibration

The wavelength calibration of CRIRES spectra is the most critical step in the whole data reduction procedure mainly because of the small number of known and/or suitable lines.

The wavelength scale can be calibrated with either arc frames or the OH lines that are imprinted on each spectrum. Normally, the advantage of the arc is that there are lines covering the entire wavelength range, although given the high resolution of CRIRES the density of the known lines is quite small. The disadvantage is that the arcs are taken separately and, in most cases this means that the grating has moved between the time when the target was observed and the arcs were taken. One can use the OH lines, if present, to cross check and correct the zero point of the wavelength calibration.

The advantage of the OH lines is that above $2.2\mu\text{m}$ they are numerous and that they may lead to a slightly more accurate wavelength calibration. The disadvantage is that below $2.2\mu\text{m}$ they are often too faint to allow a good fit.

In general, for an accuracy corresponding to ~ 3 pixels, the start and end wavelength and the derived dispersion, assumed to be linear for each detector is sufficient, in particular for the CRIRES standard settings.

It is worth mentioning that, in case the wavelength calibration is done by using ThAr lamp frame and the CRIRES pipeline (with the **crires_spec_wave** recipe), in order to improve the cross-correlation, one should run the recipe by using not only the lamp and master flat frames but also a *master dark* frame, taken with the same DIT as the lamp flat. An alternative is the use of the **crires_util_wlcalib** recipe.

In order to perform the wavelength calibration procedure with IRAF, one can use the task *identify*, under the *onedspec* environment.

5.4 Combining 2d spectra

For the classical ABBA technique, one multiplies each image by -1 and adds it back to itself after a suitable shift. This method of combining data is often called *double sky subtraction* as it effectively removes any residual sky that remains after the first subtraction. It results in an image that has one positive spectrum and two negative spectra either side of the positive spectrum.

In the more complex cases, one combines the individual spectra after suitable shifts have been applied.

To combine 2d spectra, one can either use the CRIRES pipeline (**crires_util_combine** recipe) or, with IRAF by using the task *mscarith* under *mscred* environment.

5.5 Extraction

For the classical ABBA technique, one should extract the spectrum without fitting the sky. Fitting the sky only adds noise. For more complex cases, a low order fit to the sky may be required.

As for all the steps listed above, the extraction procedure can be done with several different tools, such as IRAF, MIDAS or of course by using the CRIRES pipeline. In the latter case, running the recipe **crires_spec_jitter** will allow simultaneously, to correct the science data for non-linear effects, flat-fielding, wavelength calibration and finally extract the spectrum. An alternative is the use of the recipe **crires_util_extract**. Note that in v2.3 of the pipeline the *starty* and *stopy* parameters have been replaced by *y_pos_c1,2,3,4* and *y_width* options. For example, in place of *-start_y=100 -start_y=200* one would use recipe parameters *-y_pos_cN="150"* and *-y_width=100*.

In case one is dealing with science frames showing more than one spectrum (e.g. because there are more than one source on the slit), the CRIRES pipeline, by default, will extract only the brightest spectrum. Therefore, to extract the fainter one you should run the **crires_spec_jitter** recipe with suitable values of *starty* and *stopy*, which define the region (in pixel) where the spectrum is.

If one is using IRAF, the extraction of the spectra can be done with the task *apall*, under the *twodspec> apexextract* environment.

5.6 Removing telluric lines

This is a critical step that requires some care and, possibly, some experimentation. The aim is to remove the large number of telluric lines that appear in IR spectra. This is done by dividing the object spectrum with that of a telluric standard or a model spectrum of the sky as created by (for example):

www.eso.org/observing/etc/skycalc/skycalc.htm. Since this is a division of one spectrum by another, it is important that the strength, shape and centroid of the telluric lines match. First and foremost, the telluric standard and the object have to be observed with the same instrument setup, with roughly the same airmass and, if possible, consecutively. Secondly, the object and science data should be reduced in the same way and with the same calibration frames. For the best results, one may have to modify the spectrum of the telluric standard so that the center and strength of the telluric lines match those of the object spectrum. The next step is to remove spectral features that have been imprinted onto the object spectrum from the telluric standard itself. Telluric standards are either hot stars or solar-type dwarfs. Both types contain spectral features that should be removed. For solar-type stars, one can use the observed solar spectrum to remove the features. Hot stars usually contain helium and hydrogen lines. If the spectral regions around these lines are of interest, then one should think carefully about using these type of stars. However, one can still try to remove these lines by fitting them, or alternatively, use a second telluric standard that does not have helium or hydrogen lines so that these lines can be removed from the hot star. If required, users

should provide their own telluric standard(s). In the case where there is a problem with the provided telluric, typically an observatory standard will be taken. In these cases the CRIRES FITS header does not contain the full target name of the telluric standard and, sometimes, operators forget to include it in the night report. To find out which telluric standard was used, look at the RA and DEC of the target and consult the following link: <http://www.eso.org/sci/facilities/paranal/sciops/catsearch.html>

5.7 Flux calibration

The first step is to obtain a relative flux calibration. The second step is to do absolute flux calibration.

If the telluric standard was a hot star, then a blackbody curve can be used to model the continuum of the standard. The spectral type of the star can be used to give an idea of what temperature to use. The blackbody curve is then multiplied into the object spectrum.

For solar-type stars, a blackbody curve is a good enough description of the spectral energy distribution above 1.6 microns. Below 1.6 microns, a more accurate description of the continuum is required.

The second step is absolute flux calibration. If the magnitude of the target is known one can determine the appropriate scaling by convolving the spectrum with the filter curve. CRIRES standard star observations typically are done with a 0.4 arcsecond slit, hence slit losses are an issue even when using adaptive optics. In the case the telluric has been observed with the same slit width, it is possible to estimate the slit losses by taking a "cross cut" across the spatial axis for both science and telluric spectra. Alternatively, if the seeing did not change too drastically while the target and the standard were observed, slit losses can be ignored. Very accurate flux calibration is difficult with CRIRES.

6 The ESO CRIRES pipeline: installation

The ESO CRIRES pipeline can be downloaded at <http://www.eso.org/sci/software/pipelines/>, where you will find the release file (including CPL, Gasgano, EsoRex and calibration data) plus the excellent pipeline manual.

Note that many of the examples in this cookbook refer to version 2.1.3 of the pipeline, however, the newest version should always be used (which at the time of the release of this document is 2.3.1).

Once downloaded, you simply have to proceed as follows:

```
$ pwd
/home2/jsmoker/Instruments/CRIRES/Pipeline
$ gunzip crire-kit-2.1.3.tar.gz

$ tar xvf crire-kit-2.1.3.tar
cd crire-kit-2.1.3
sh install_pipeline
I am about to install the following software packages:
  gasgano-2.4.0.tar.gz
  cfitsio3090.tar.gz
  cpl-5.3.1.tar.gz
  esorex-3.9.0.tar.gz
  crire-2.1.3.tar.gz
- with the following pipeline calibration file(s):
  crire-calib-2.1.3.tar.gz
```

The software installation will be organised in bin/, lib/, include/ and crire-2.1.3/ directories, while the calibration files will be installed in a calib/ directory.

Oops, I cannot proceed with the gasgano installation:

```
gasgano must be installed in /home/jsmoker/gasgano.
You appear to already have gasgano installed there.
I intended to rename your existing gasgano installation to
/home/jsmoker/gasgano.old but
/home/jsmoker/gasgano.old already exists.
```

You can either let me

D) Delete the current /home/jsmoker/gasgano before I install gasgano there,
R) Replace /home/jsmoker/gasgano.old with your current /home/jsmoker/gasgano, or
S) Stop, so you can backup both /home/jsmoker/gasgano and /home/jsmoker/gasgano.old,
and then restart the installation.

D) Delete, R) Replace, S) Stop ? [d/R/s] R

Where should I install the software packages ? [/home/jsmoker]

/home/jsmoker/Instruments/CRIRES/Pipeline/crire-kit-2.1.3/

Where should I install the pipeline calibration files ?

```
[/home/jsmoker/Instruments/CRIRES/Pipeline/crre-kit-2.1.3/]
```

<...snip...>

Updating /home/jsmoker/gasgano/config/CRIRE.prefs

Updating /home/jsmoker/.gasganorc

In order to use the recipes from gasgano you need to set the environment variable CPLDIR to /home/jsmoker/Instruments/CRIRES/Pipeline .

Depending on your type of shell a possible command for doing this may be:
export CPLDIR=/home/jsmoker/Instruments/CRIRES/Pipeline

Warning:

Your PATH does not point to your newly installed
/home/jsmoker/Instruments/CRIRES/Pipeline/bin/esorex and
/home/jsmoker/gasgano/bin/gasgano.

To ensure execution of the newly installed software you should modify your PATH.

Depending on your type of shell a possible command for doing this may be:
export PATH=/home/jsmoker/starlink/star_new_hawaiki/star/java/jre/bin:
/scisoft/bin:
/home/jsmoker/gasgano/bin:/home/jsmoker/Instruments/CRIRES/Pipeline/bin

In the above case I use tcsh so define the following aliases in my .tcshrc before using the pipeline:

```
$ alias crires_cpl "setenv CPLDIR /home/jsmoker/Instruments/CRIRES
                    /Pipeline/crre-kit-2.1.3"
$ alias crires_pipe "setenv PATH /home/jsmoker/Instruments/CRIRES/Pipeline
                    /crre-kit-2.1.3/bin:/home/jsmoker/gasgano/bin:/home/jsmoker/java/jdk_16
                    /jdk1.6.0_16/bin:/home/jsmoker/Instruments/CRIRES/Pipeline/crre-kit-2.1.3
                    /crre-2.1.3/recipes/.libs/:${PATH};setenv LD_LIBRARY_PATH
                    ${LD_LIBRARY_PATH}:/home2/jsmoker/Instruments/CRIRES/Pipeline
                    /crre-kit-2.1.3/lib"
```

6.1 Opening gasgano, resizing columns and selecting data for reduction

Remember that a full description of gasgano can be found here: <http://www.eso.org/sci/software/gasgano/>. First we want to make sure that we are looking in the correct place for the pipeline, java version and gasgano.

Using these aliases defined above we setup the paths and run gasgano:

```
$ crires_cpl
$ crires_pipe
$ gasgano
```

At this point gasgano should fire up and if you click on the small tabs within it, you should see a window as shown in Fig. 17, with lines in red such as "lines_hitran.fits, lines_model.fits" and so on. These are calibration files that may be used in the data reduction.

Note that to see the headers better (for example "CLASSIFI"), then move the mouse to the right hand side of the column in question until the left/right arrow appears, then left mouse click and drag the column heading to the right to increase its size.

Now we want to make it possible for gasgano to see the raw data:

- File → Add/Remove file → Add file → (in this case /home/jsmoker/Instruments/CRIRES/Pipeline/Data/Raw and Reduced) then OK (see Fig 18). At this point the CRIRES raw files are visible in blue.

With the raw files selected we are now ready to perform data reduction using the pipeline.

6.2 The ESO CRIRES pipeline under MacOS

The ESO CRIRES pipeline does not support MAC OS and the quality of the results produced has not been verified. There may be subtle effects in the pipeline output that are not immediately obvious. The author uses linux so has not been able to check the following, so the information is given "as is".

To compile the pipeline using 64 bits 10.6.8, java 1.6 then using csh one may try:

```
bash
export JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home
./install pipeline
follow the instructions at the end of the installation to set
properly the gasgano/esorex paths and the environment variables
(setenv in csh, export in bash)
```

The information above was taken from <http://www.sc.eso.org/~epompei/mac.html>

If one tries to launch gasgano after having installed the pipeline on a MAC, gasgano will fail because there are some definitions in the gasgano script which do not work; in particular the path for the CPLdir is incorrect. In order to correct this the user should first install the pipeline as described in the web pages, then before launching gasgano the file called 'gasgano' (which is under /home/gasgano/bin) needs to be edited as follows:

- Comment out the following lines

```
DYLIB_LIBRARY_PATH=$CPLDIR/lib:$BASE/lib:$DYLIB_LIBRARY_PATH
export DYLIB_LIBRARY_PATH
optjre="$optjre -Djava.library.path=$DYLIB_LIBRARY_PATH"
```


- Replace them with the following:

```
DYLD_LIBRARY_PATH=$CPLDIR/lib:$BASE/lib:$DYLD_LIBRARY_PATH
export DYLD_LIBRARY_PATH
optjre="$optjre -Djava.library.path=$DYLD_LIBRARY_PATH"
```

6.3 The ESO CRIRES pipeline reflex environment

In a future release it is hoped that data reduction of CRIRES data will be possible using the reflex environment which allows more "hands on" interaction with the reduction process.

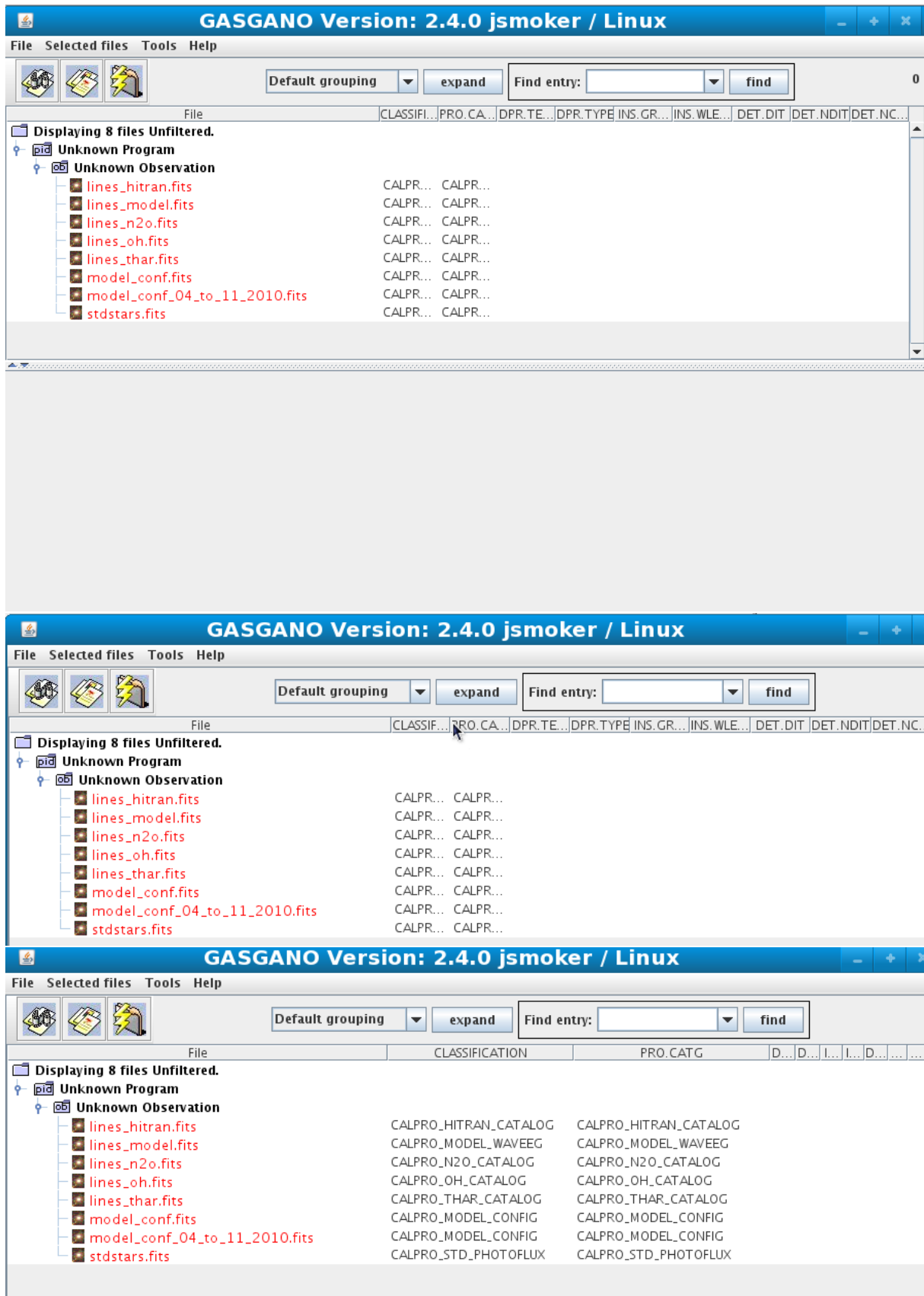


Figure 17: Top panel: Initial opening screen for gasgano. Middle panel: Before increasing the column width. Bottom panel: Column width in gasgano increased.

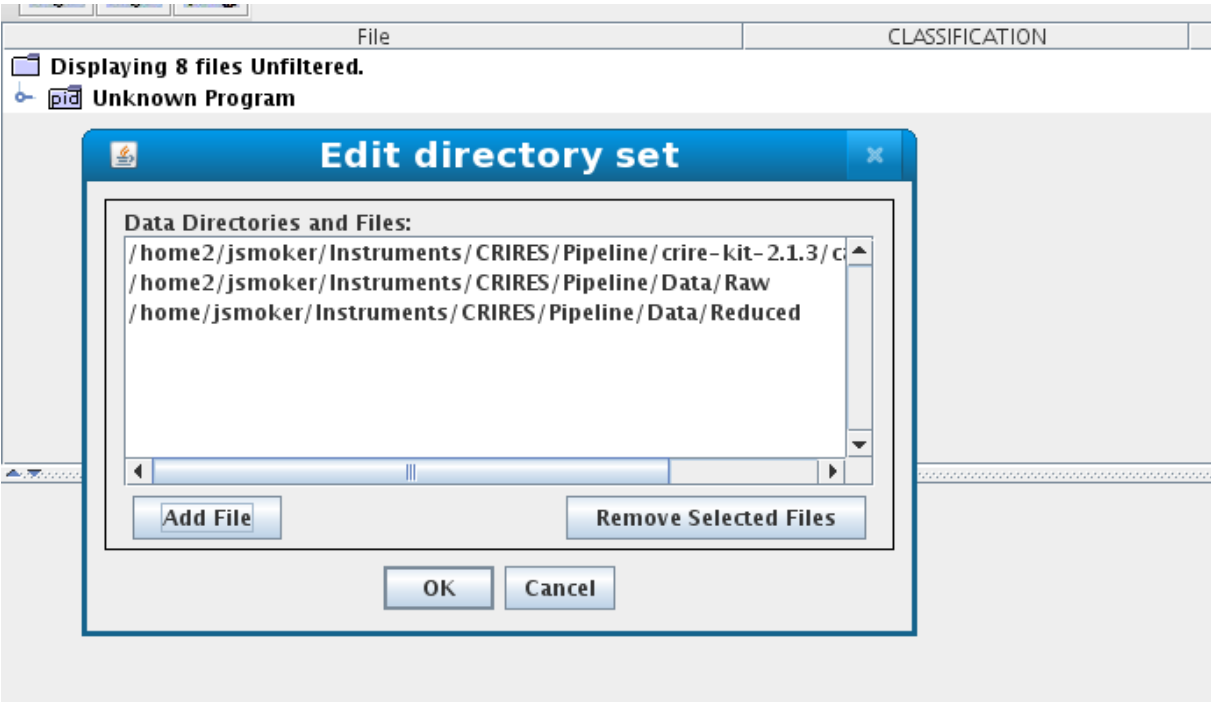


Figure 18: Adding files in gasgano.

7 The ESO CRIRES pipeline: Quick look data reduction of CRIRES nodding and stare images using the recipe `crires_spec_jitter`

We will describe the data reduction of a standard star at two wavelengths, 1087.3-nm and 3611.0-nm, using both `gasgano` and `esorex`. The first example will be of spectra taken by nodding along the slit, finally an example of a quick-look reduction of a stare image will be presented.

Your first quick reduction will use just two scientific spectra (one in each nodding position) and the recipe `crires_spec_jitter`.

7.1 Data reduction of nodding spectra using `gasgano`

Within `gasgano` we click on the file or files that we wish to reduce (in this case CRIRES.2010-10-20T03:28:13.918 which is a spectrum of wavelength 3611.00-nm. Then go to "Selected Files" → "To Recipe" → "Load Recipe" and `crires_spec_jitter` (Fig. 19).

At this point a further window will appear (`crires_spec_jitter_v20103`). Going from top to bottom there are several sub-windows, which are explained as follows.

- Parameters. For the `crires_spec_jitter` recipe there are 23 parameters that you can alter as you please in order to tune the data reduction (top panel of Fig. 20).
- Input frames. In this case we have only (incorrectly...) added one input frame. You will notice that `gasgano` has classified this frame as CAL_NOD, using the classification rules. Note that if there is a problem with the classification or you want to fool the system, you can **edit** this field.
- Product naming. By default the products are placed in your home directory. It is advisable to change this to somewhere else (in this case /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced to prevent large number of files being present in your home area. Note that the **Naming Scheme** can be set to Numeric or Overwrite. In the first case whenever you run a reduction it will be named sequentially, which can be useful when comparing reductions done with different parameters. Alternatively with Overwrite only the latest reduction will be saved.
- Output frames.
- Log Messages. These are also saved in ASCII files in the products directory with names such as `crires_spec_jitter_2011-09-29_10:41:55.log`

OK, now it is time to run the first reduction. Hit the `Execute` button and look at Fig. 20 for the output.

It **failed!** Why? Well, if we look in the error log we see the following message:

```
<snip>
ERROR: 11:57:39 [ ERROR ]           Only one offset in the list - abort
ERROR: 11:57:39 [ ERROR ]           Cannot classify the data
```

```
ERROR: 11:57:39 [ ERROR ]      [1/1] 'Illegal input' (14) at
                                crires_spec_jitter:crires_spec_jitter.c:510
Completion status: FAILURE
Execution error: Execution failed with code 14
```

So, the recipe had problems because we only gave it one nodding position. Let's add the other nodding position ("Selected Files" → "To Recipe" → `crires_spec_jitter_v20103`. This will then appear in the "Input Frames" sub-window. We now have a full A-B pair, so let's try again to run the recipe. Hit and wait for a while until we see "SUCCESS" (Fig. 21). The reduced files are visible in red (see Fig. 22). Note that one *can* use the recipe with only one nodding position, but the input parameters have to be changed (e.g. see Sect. 7.3).

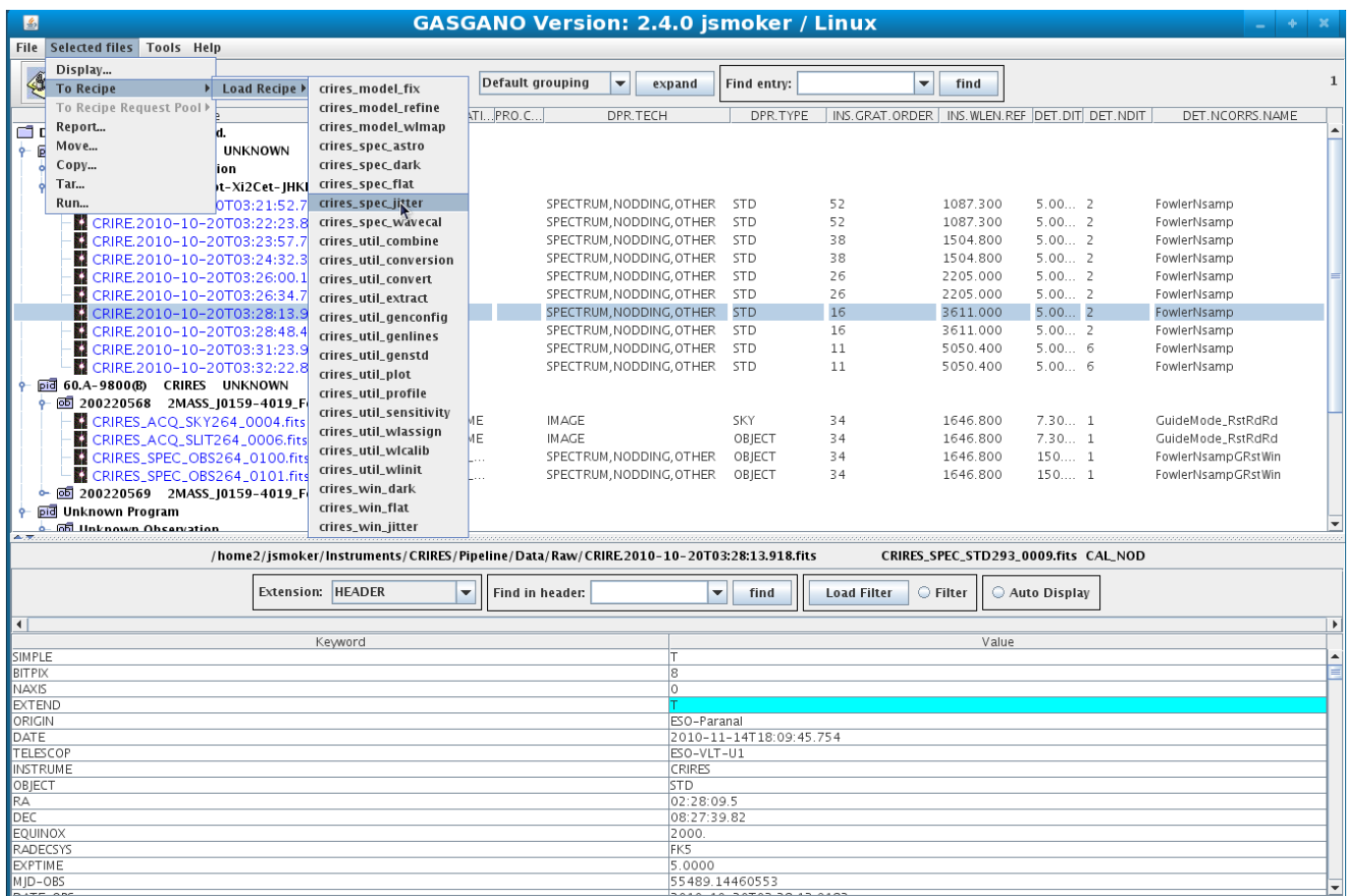


Figure 19: Preparing the reduction of nodding observations in gasgano using the recipe CRIRES_SPEC_JITTER.

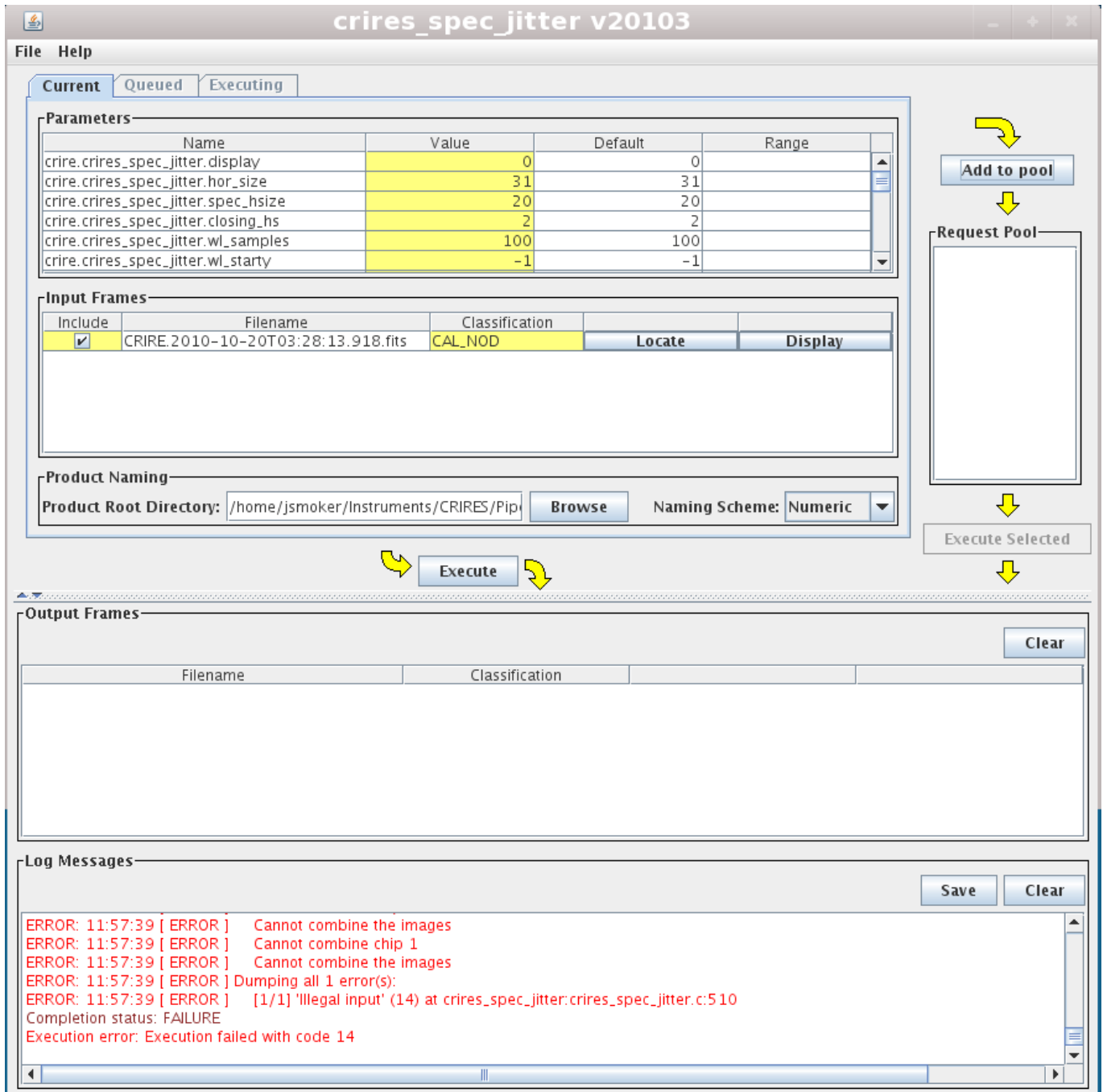


Figure 20: Failed reduction because only one spectrum was used in the recipe `crir. crires_spec_jitter` with the default parameters.

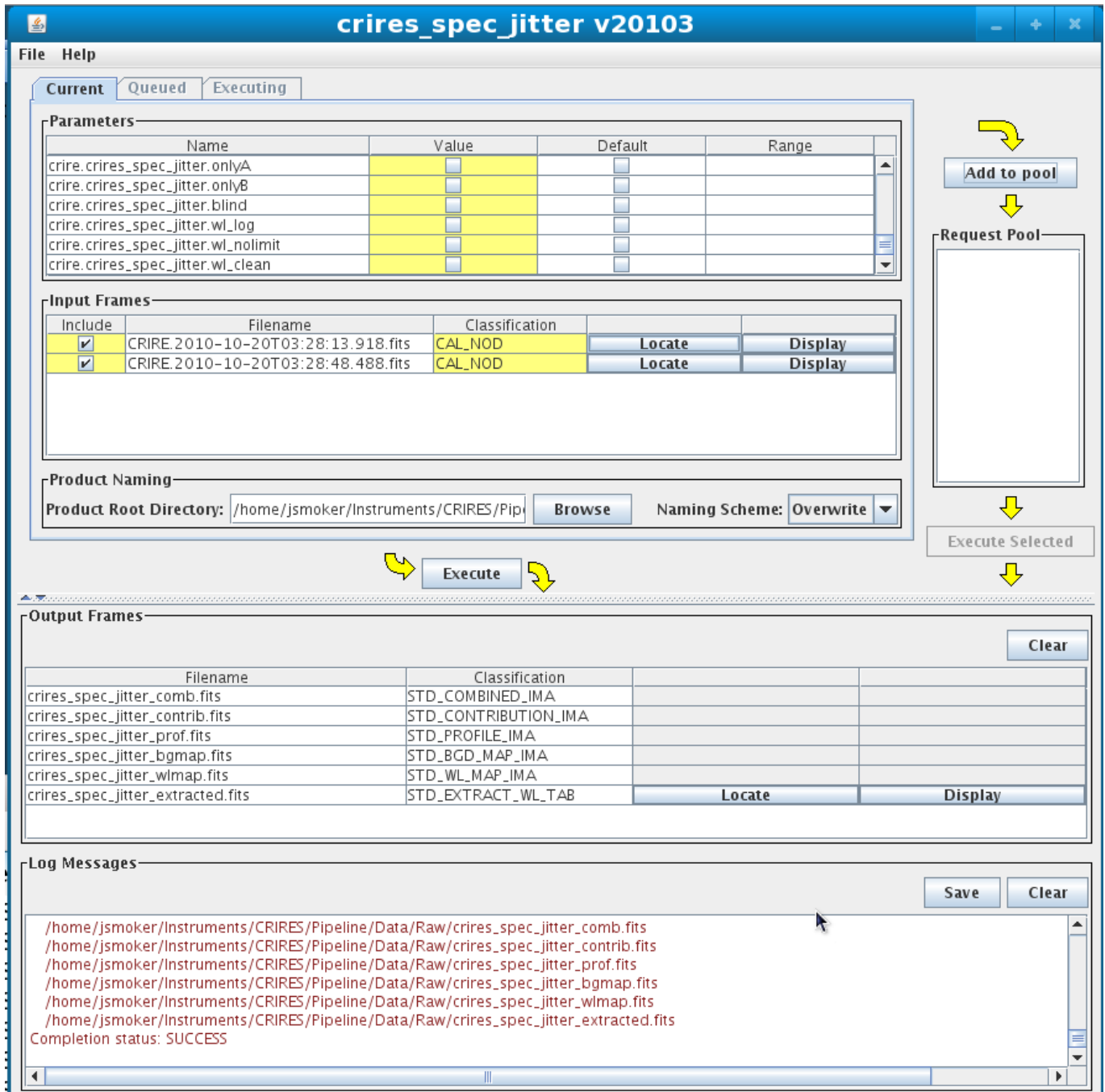


Figure 21: Successful reduction using the `crirspec_jitter` recipe. Note that this is the simplest reduction possible; for example there is NO correction for the odd-even effect, flatfielding, or good wavelength calibration.

GASGANO Version: 2.4.0 jsnoker / Linux

File Selected files Tools Help

Default grouping expand Find entry: find

Displaying 55 files Unfiltered.

File	CLASSIFICATION	PRO. CATG	DPR. TECH	DPR. TYPE	INS. GRAT. O...	INS. WLEN...	DET. DIT	DET. NDIT	DET. NCORRS. NA...
60.A-9051(A) CRIRES UNKNOWN									
200163466 Calibration									
200165585 specphot-Xi2Cet-JHKLM									
CRIRES_2010-10-20T03:21:52.798.fits	CAL_NOD		SPECTRUM,NODDI...	STD	52	1087.300	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:22:23.810.fits	CAL_NOD		SPECTRUM,NODDI...	STD	52	1087.300	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:23:57.755.fits	CAL_NOD		SPECTRUM,NODDI...	STD	38	1504.800	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:24:32.399.fits	CAL_NOD		SPECTRUM,NODDI...	STD	38	1504.800	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:26:00.136.fits	CAL_NOD		SPECTRUM,NODDI...	STD	26	2205.000	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:26:34.717.fits	CAL_NOD		SPECTRUM,NODDI...	STD	26	2205.000	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:28:13.918.fits	CAL_NOD		SPECTRUM,NODDI...	STD	16	3611.000	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:28:48.488.fits	CAL_NOD		SPECTRUM,NODDI...	STD	16	3611.000	5.00...	2	FowlerNsamp
CRIRES_2010-10-20T03:31:23.984.fits	CAL_NOD		SPECTRUM,NODDI...	STD	11	5050.400	5.00...	6	FowlerNsamp
CRIRES_2010-10-20T03:32:22.834.fits	CAL_NOD		SPECTRUM,NODDI...	STD	11	5050.400	5.00...	6	FowlerNsamp
crires_spec_jitter_bgmap.fits	STD_BGD_MAP_IMA	STD_BGD_MAP_IMA			16	3611.	5.	2	FowlerNsamp
crires_spec_jitter_comb.fits	STD_COMBINED_IMA	STD_COMBINED_IMA			16	3611.	5.	2	FowlerNsamp
crires_spec_jitter_contrib.fits	STD_CONTRIBUTION_IMA	STD_CONTRIBUTION_IMA			16	3611.	5.	2	FowlerNsamp
crires_spec_jitter_extracted.fits	STD_EXTRACT_WL_TAB	STD_EXTRACT_WL_TAB			16	3611.	5.	2	FowlerNsamp
crires_spec_jitter_prof.fits	STD_PROFILE_IMA	STD_PROFILE_IMA			16	3611.	5.	2	FowlerNsamp
crires_spec_jitter_wlmap.fits	STD_WL_MAP_IMA	STD_WL_MAP_IMA			16	3611.	5.	2	FowlerNsamp
60.A-9800(B) CRIRES UNKNOWN									
200220568 2MASS J0159-4019.FowlerNsampGRs									
CRIRES_ACO_SKY264_0004.fits	ACO_FRAME		IMA_CFE	SKY	34	1646.800	7.30	1	GuidedMode_Rst

crires_spec_jitter_extracted.fits <unknown orig name> STD_EXTRACT_WL_TAB

Extension: HEADER Find in header: find Load Filter Filter Auto Display

Keyword	Value
SIMPLE	T
BITPIX	16
NAXIS	0
EXTEND	T
DATE	2011-09-29T16:20:34
ORIGIN	ESO-Paranal
TELESCOP	ESO-VLT-U1
INSTRUME	CRIRES
OBJECT	STD
RA	02:28:09.5
DEC	08:27:39.82
EQUINOX	2000.
RADECSYS	FK5
MJD-OBS	55489.14460553
DATE-OBS	2010-10-20T03:28:13.9182

Figure 22: Gasgano raw and reduced data obtained with `crires_spec_jitter`.

7.2 Quick data reduction of nodding spectra using esorex

??

For the simplest data reduction of nod/jitter observations using ESOREX, one needs only the following .sof file:

```
$ cat Science_Frames_1087p3.sof
CRIRE.2010-10-20T03:21:52.798.fits CAL_NOD
CRIRE.2010-10-20T03:22:23.810.fits CAL_NOD
```

And then run the following command:

```
$ esorex --output-dir=/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced
          crires_spec_jitter Science_Frames_1087p3.sof
$ mv out_0000.fits std_combined_ima_1087p3.fits
$ mv out_0001.fits std_contribution_ima_1087p3.fits
$ mv out_0002.fits std_profile_ima_1087p3.fits
$ mv out_0003.fits std_bgd_map_ima_1087p3.fits
$ mv out_0004.fits std_wl_map_ima_1087p3.fits
$ mv out_0005.fits std_extract_wl_tab_1087p3.fits
```

...and run something similar for the other wavelength setting with a different .sof file. Note that these data are uncalibrated.

7.3 Quick reduction of stare frames using esorex

One can reduce staring frames using the pipeline with the use of the "blind" parameter within the `crires_spec_jitter` recipe. For example, if we have a single stellar observation as shown in Fig. 25, we create a simple .sof file like:

```
% cat QuickReduceNoCals_HD29085_OBS_DIR_NoNod.sof
GasCell_HD29085_1579p137_272_0009.fits OBS_DIR
```

....and reduce it in esorex using the following command:

```
esorex --recipe-dir=/path/crire-kit-2.3.0/crire-2.3.0/recipes/.libs
       --output-dir=/path/Projects/NoNod crires_spec_jitter
       --y_pos_c1="290" --y_pos_c2="289" --y_pos_c3="274"
       --y_pos_c4="264" --y_width=30
       --blind=true --display=3 QuickReduceNoCals_HD29085_OBS_DIR_NoNod.sof
```

Note that "y_pos_c1,2,3,4" is a list of Y positions on detectors 1-4 where to perform the wavelength calibrations (default is to use the spectrum position)". This is **not** where the spectrum is extracted. In order to extract at a particular Y position you must use the recipe parameter "spec_zone".

Note that one can also "extract" non-stellar spectra in the same way (for example gas-cell flatfields etc), although **sometimes** the OPTimal extractions will fail on detector two as there is no profile to optimally extract. In this case just inspect the RECTangular extractions. In the example shown below, both OPT and RECT extraction worked on the raw data shown in Fig. 24 with .sof file Gas_Cell_No_Star_RawData.sof:

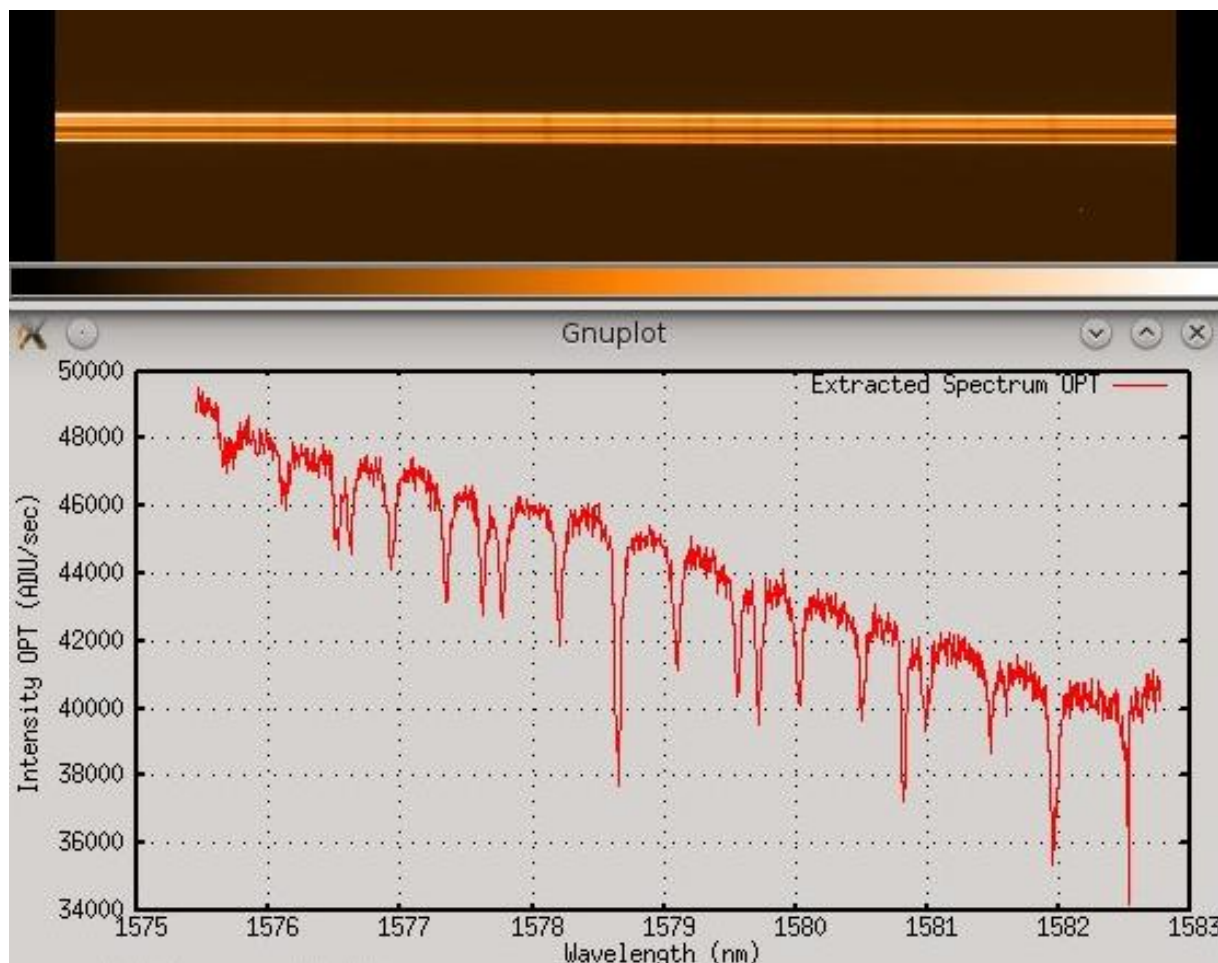


Figure 23: Example of a CRISTES stare observation and quick-look reduction using `crires_spec_jitter` and the `blind="true"` option. Detector 3 only is shown.

```
% cat Gas_Cell_No_Star_RawData.sof
Gas_Cell_No_Star_RawData.fits OBS_DIR
```

...now run esorex:

```
esorex --recipe-dir=/path/crire-kit-2.3.0/crire-2.3.0/recipes/.libs
--output-dir=/path/Projects/NoNod crires_spec_jitter
--y_pos_c1="290" --y_pos_c2="289" --y_pos_c3="274"
--y_pos_c4="264" --y_width=30
--blind=true --display=3 Gas_Cell_No_Star_RawData.sof
```

..to produce the following output:

```
***** ESO Recipe Execution Tool, version 3.9.6 *****
[ INFO ] crires_spec_jitter: [tid=000] *****
[ INFO ] crires_spec_jitter: [tid=000] Detector 1 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: [tid=000] Detector 2 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: [tid=000] Detector 3 : Y=1 => Y=512 pixels
```

```

[ INFO ] crires_spec_jitter: [tid=000] Detector 4 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: [tid=000] *****
<snip>
[WARNING] crires_spec_jitter: [tid=000]   The spectrum positions in chip 2
                                     and chip 3 are too different: 220 -> 189
[WARNING] crires_spec_jitter: [tid=000]   The spectrum positions in chip 3
                                     and chip 4 are too different: 189 -> 328
<snip>
[ INFO ] crires_spec_jitter: [tid=000]   Save the product
[ INFO ] crires_spec_jitter: [tid=000]   Writing FITS propertylist
      product(OBS_COMBINED_IMA): crires_spec_jitter_comb.fits
[ INFO ] crires_spec_jitter: [tid=000]   Writing FITS propertylist
      product(OBS_CONTRIBUTION_IMA): crires_spec_jitter_contrib.fits
[ INFO ] crires_spec_jitter: [tid=000]   Writing FITS propertylist
      product(OBS_PROFILE_IMA): crires_spec_jitter_prof.fits
[ INFO ] crires_spec_jitter: [tid=000]   Writing FITS propertylist
      product(OBS_BGD_MAP_IMA): crires_spec_jitter_bgmap.fits
[ INFO ] crires_spec_jitter: [tid=000]   Writing FITS propertylist
      product(OBS_WL_MAP_IMA): crires_spec_jitter_wlmap.fits
[ INFO ] crires_spec_jitter: [tid=000]   Writing FITS table
      product(OBS_EXTRACT_WL_TAB): crires_spec_jitter_extracted.fits
[ INFO ] esorex: [tid=000]   Created product
      /diska/home/jsmoker/CRIRES/Projects/NoNod/out_0000.fits
[ INFO ] esorex: [tid=000]   Created product
      /diska/home/jsmoker/CRIRES/Projects/NoNod/out_0001.fits
[ INFO ] esorex: [tid=000]   Created product
      /diska/home/jsmoker/CRIRES/Projects/NoNod/out_0002.fits
[ INFO ] esorex: [tid=000]   Created product
      /diska/home/jsmoker/CRIRES/Projects/NoNod/out_0003.fits
[ INFO ] esorex: [tid=000]   Created product
      /diska/home/jsmoker/CRIRES/Projects/NoNod/out_0004.fits
[ INFO ] esorex: [tid=000]   Created product
      /diska/home/jsmoker/CRIRES/Projects/NoNod/out_0005.fits
[ INFO ] esorex: [tid=000]   6 products created
jsmoker@pc015042 NoNod]$

```

The reduced spectra for this (uncalibrated detector 3 only) extraction are shown in Fig. 25.

7.4 Plotting the reduced spectra

We see that the reduced products are now visible in gasgano in RED (Fig. 22), with names `crires_spec_jitter*fits`. To display them we select as input `STD_EXTRACT_WL_TAB` and send them to the recipe `crires_util_plot`. By default, 8 plots will be produced. In order to remove unwanted plots from the screen, hit 'q'. Figure 26 shows the last window plotted, which shows the final extracted spectrum obtained using optimal extraction.



Figure 24: Raw detector 2 and 3 only images of a gas-cell.

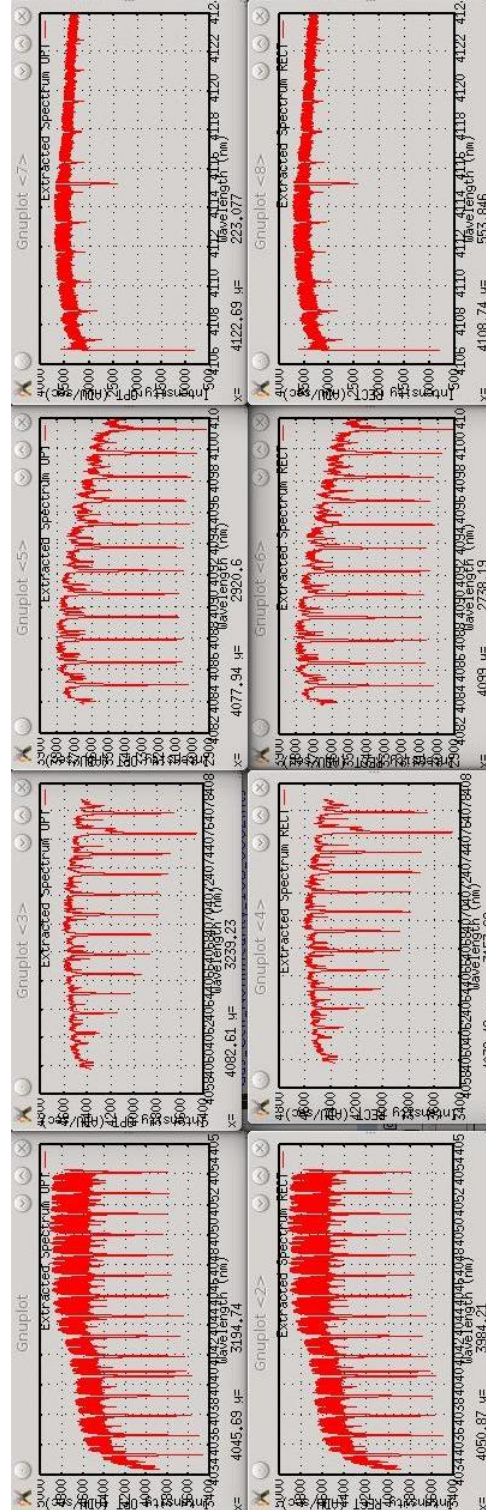


Figure 25: Reduced OPTimal and RECTangular extraction of all four detectors for the gas-cell data shown in Fig. 24.

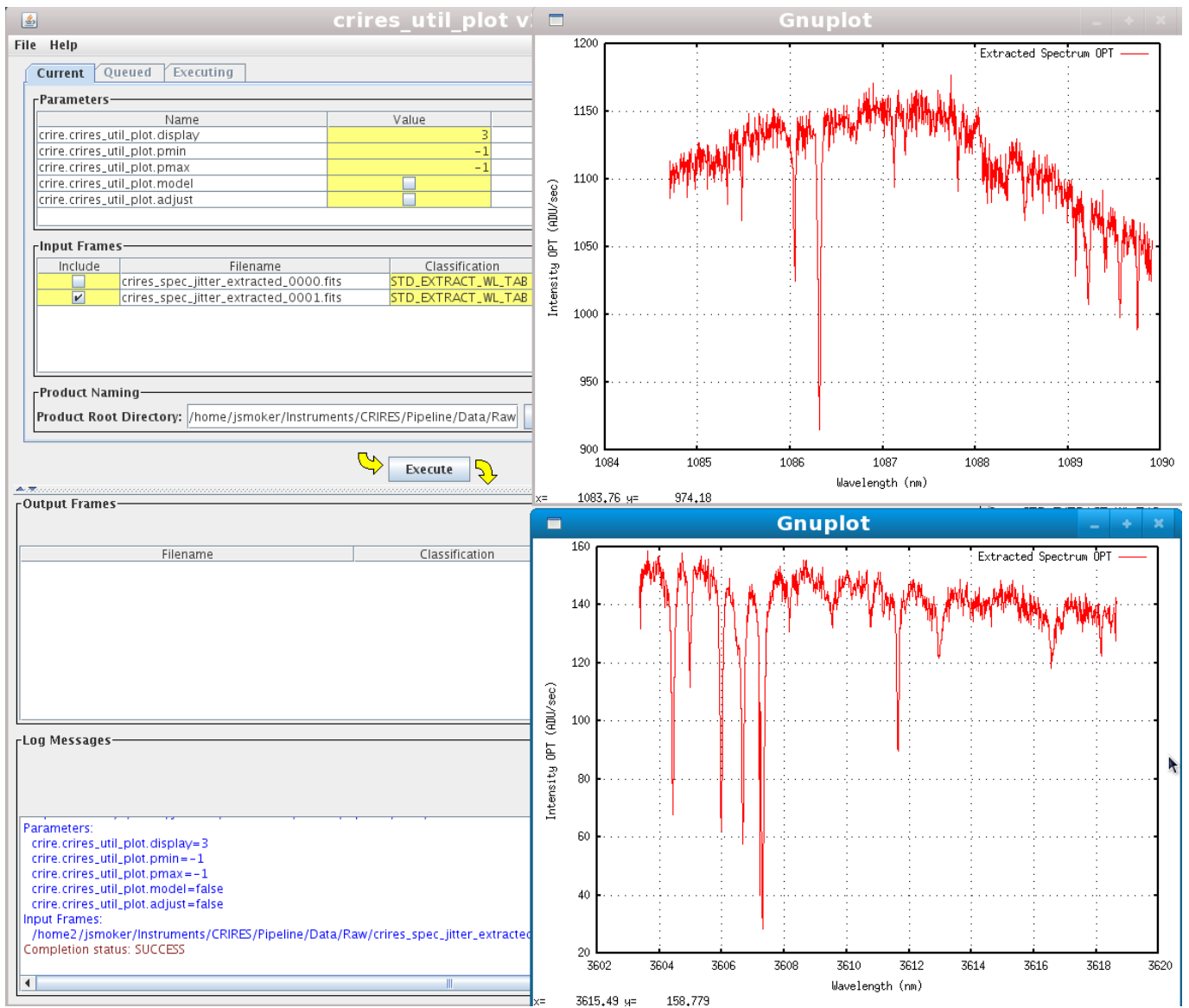


Figure 26: Plot of reduced spectra (detector 3) using with the `crires_util_plot` recipe. The recipe was run twice to display the two different wavelength ranges for the two quick reductions at 1087.3 and 3711.0-nm.

8 The ESO CRIRES pipeline: Full data reduction of CRIRES nodding images

The full data cascade of CRIRES is described in Sect. 8 of the pipeline manual and is followed here for the nodding case. Note that in the majority of cases the output from one recipe (e.g. `crires_spec_dark`) is used by the next recipe in the chain (e.g. `crires_spec_flat`).

- Dark creation: `CAL_DARK` → `crires_spec_dark`
- Linearity: `ON_RAW` & `OFF_RAW` → `detmon_ir_lg`. Available soon. At the time of writing users should download data from:
http://www.eso.org/observing/dfo/quality/CRIRES/pipeline/pipe_calib.html.
- Flatfielding: `CAL_FLAT` → `crires_spec_flat`
- Wavelength calibration: `CAL_WLSKY/CAL_WLLAMP/CAL_WLABS` → `crires_spec_wavecal`
- Wavelength refinement: `MODEL_REFINE_RAW` → `crires_model_refine`
- Final reduction: `OBS_DIR` or `DIR_JIT/OBS_NOD` or `NOD_JIT/CAL_DIR` or `DIR_JIT/CAL_NOD` or `NOD_JIT/OBS_OBJECT` or `SKY/OBS_GENERIC_OFFSET` or `SKY` → `crires_spec_jitter`

We will concentrate on two cases, a standard star with a reference wavelength at 1087.3-nm and the same bright standard at 3611.0-nm.

8.1 Look at the raw data and calibrations

One should use `dfits` to see what data are available and then use programmes such as `rtd` or `ds9` and assess their quality. In the current case we have standard star observations, flats and ThAr wavelength calibrations taken at 1087.3-nm and standard star and flatfields only available for the 3611.0-nm; remember that no ThAr exposures are taken redwards of 2400-nm.

```
$ dfits CRI*fits | fitsort OBJECT INS.WLEN.REF DET.DIT DPR.CATG DPR.TYPE | grep 1087
```

CRIRE.2010-10-20T03:21:52.798.fits	STD	1087.300	5.0000000	CALIB
CRIRE.2010-10-20T03:22:23.810.fits	STD	1087.300	5.0000000	CALIB
CRIRE.2010-10-20T10:00:53.435.fits	FLAT	1087.300	5.0000000	CALIB
CRIRE.2010-10-20T10:02:18.593.fits	FLAT	1087.300	5.0000000	CALIB
CRIRE.2010-10-20T10:03:43.485.fits	FLAT	1087.300	5.0000000	CALIB
CRIRE.2010-10-20T13:59:48.752.fits	WAVE,LAMP	1087.300	3.0000000	CALIB
CRIRE.2010-10-20T14:01:20.131.fits	WAVE,LAMP	1087.300	3.0000000	CALIB
CRIRE.2010-10-20T14:02:47.227.fits	WAVE,LAMP	1087.300	3.0000000	CALIB

```
$ dfits CRI*fits | fitsort OBJECT INS.WLEN.REF DET.DIT DPR.CATG DPR.TYPE | grep 3611
```

CRIRE.2010-10-20T03:28:13.918.fits	STD	3611.000	5.0000000	CALIB
CRIRE.2010-10-20T03:28:48.488.fits	STD	3611.000	5.0000000	CALIB
CRIRE.2010-10-20T10:45:49.238.fits	FLAT	3611.000	5.0000000	CALIB
CRIRE.2010-10-20T10:47:14.067.fits	FLAT	3611.000	5.0000000	CALIB
CRIRE.2010-10-20T10:48:34.697.fits	FLAT	3611.000	5.0000000	CALIB

Additionally we want to use darks taken with a 5.0-s DIT to reduce the flatfields and with a 3.0-s DIT to reduce the arcs.

```
$ dfits CRI*fits | fitsort OBJECT INS.WLEN.REF DET.DIT DPR.CATG DPR.TYPE | grep DARK
CRIRE.2010-10-19T15:38:05.389.fits DARK          2308.000      3.0000000      CALIB
CRIRE.2010-10-19T15:38:25.378.fits DARK          2308.000      3.0000000      CALIB
CRIRE.2010-10-19T15:38:47.742.fits DARK          2308.000      3.0000000      CALIB
CRIRE.2010-10-20T13:56:57.981.fits DARK          5219.100      5.0000000      CALIB
CRIRE.2010-10-20T13:57:24.036.fits DARK          5219.100      5.0000000      CALIB
CRIRE.2010-10-20T13:57:52.040.fits DARK          5219.100      5.0000000      CALIB
```

Finally, we want to download the following three files that contain the non-linearity coefficients:

```
$ dfits M*fits | fitsort OBJECT INS.WLEN.REF DET.DIT DPR.CATG DPR.TYPE PRO.CATG

M.CRIRES.2010-09-21T17:04:57.320.fits FLAT,LAMP,DETCHECK 4007.          1.
M.CRIRES.2010-09-21T17:05:07.420.fits FLAT,LAMP,DETCHECK 4007.          1.
M.CRIRES.2010-09-21T17:05:13.214.fits FLAT,LAMP,DETCHECK 4007.          1.
```

Note that the coefficients **change with time** so you should check the following webpage for the correct version corresponding to the date of your observation;

http://www.eso.org/observing/dfo/quality/CRIRES/pipeline/pipe_calib.html.

Finally, note that post-April 2011 instead of DETLIN_A, _B and _C the pipeline uses a single file with category COEFFS_CUBE for the non-linearity correction.

To repeat: If you are looking at pre-April-2011 CRIRES data then you need ONE file with data category COEFFS_CUBE. If you are looking at post-April-2011 CRIRES data then you need THREE files, DETLIN_A, DETLIN_B, and DETLIN_C. These files should be included within your reduction blocks to correct for non-linearity.

8.2 Master dark creation

The first step in the creation of the darks is to find out what the DIT is of your science, standard and flatfield frames. In this case we are lucky because the standard star at 1087.3-nm has the same DIT as the flatfield i.e. 5 seconds. Hence we can use the same master dark for both images. However, the DIT used for the ThAr images is 3 seconds, so a separate dark must be used for their reduction.

```
dfits CRIRE*fits | fitsort DET.DIT DET.NDIT DPR.CATG DPR.TYPE
```

(see Sect. 8.1 for details).

The master dark is a simple recipe to run that just needs as input three raw darks, taken with the same DIT as the science. Note that in ESOs calibration plan the NDIT may be different to the science observations. As noted above, darks may also be used as input to reduce flatfield frames.

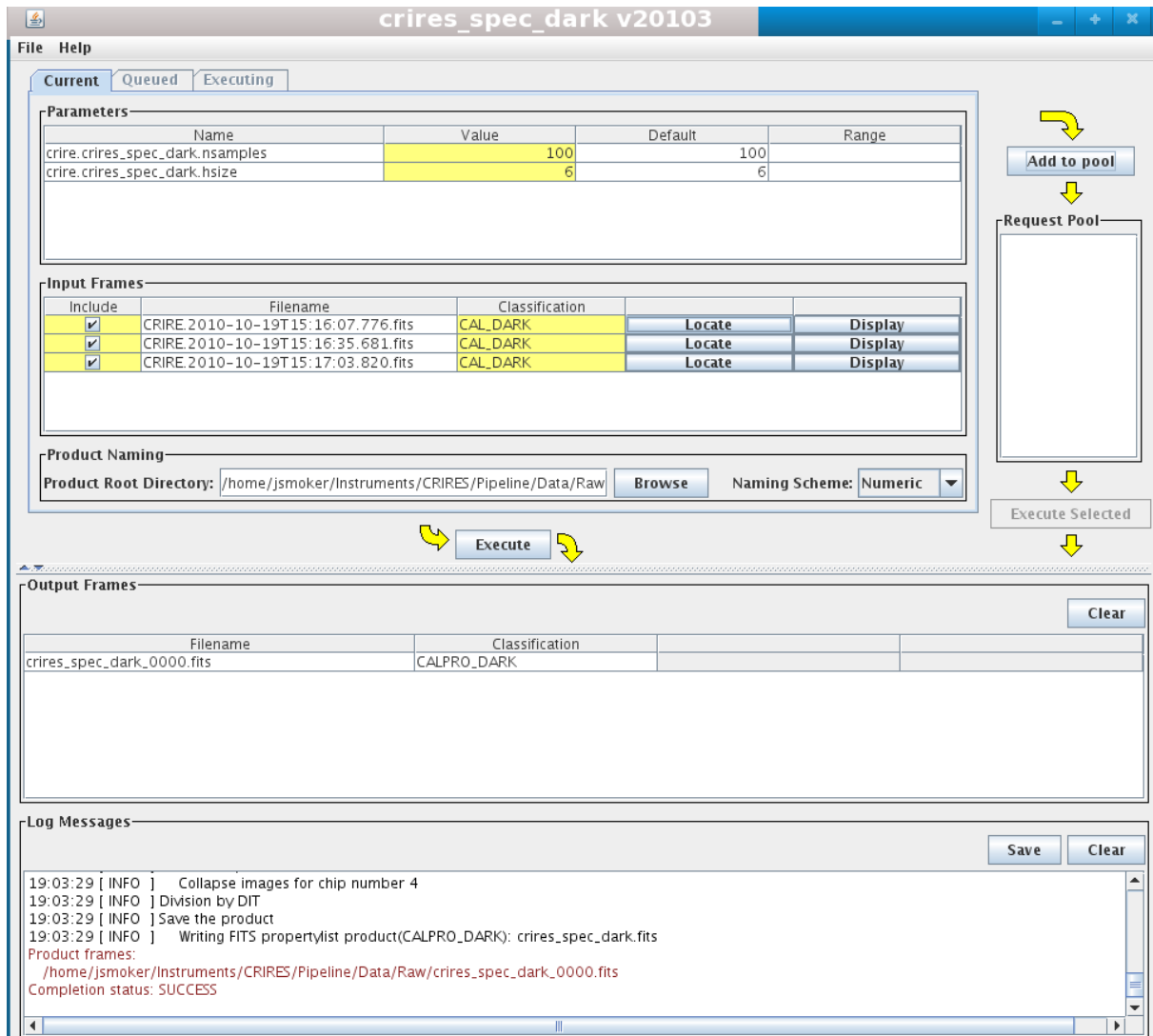


Figure 27: Master dark creation within gasgano.

8.2.1 Reduction of darks within gasgano

Figures 27 and 28 show how gasgano is used to create the master dark image, and the resulting output image. Check that it has similar counts but reduced noise to the individual dark frames. The quality control parameters are listed in the reduced image and have names like QC.* (e.g. QC.RON1, QC.DARKMED etc).

CRIRE.2010-10-20T13:56:57.981.fits	CAL_DARK	IMAGE	DARK	11	5219.100	5.000...	3	FowlerNsamp
CRIRE.2010-10-20T13:57:24.036.fits	CAL_DARK	IMAGE	DARK	11	5219.100	5.000...	3	FowlerNsamp
CRIRE.2010-10-20T13:57:52.040.fits	CAL_DARK	IMAGE	DARK	11	5219.100	5.000...	3	FowlerNsamp
CRIRE.2010-10-20T13:59:48.752.fits	CAL_WLLAMP	SPECTRUM	WAVE,LAMP	52	1087.300	3.000...	19	FowlerNsamp
CRIRE.2010-10-20T14:01:20.131.fits	CAL_WLLAMP	SPECTRUM	WAVE,LAMP	52	1087.300	3.000...	19	FowlerNsamp
CRIRE.2010-10-20T14:02:47.227.fits	CAL_WLLAMP	SPECTRUM	WAVE,LAMP	52	1087.300	3.000...	19	FowlerNsamp
crires_spec_dark_0000.fits	CALPRO_DARK	CALPRO_DARK		11	5050.4	5.	3	FowlerNsamp

200165585 specphot-X12Cet-JHKLM

crires_spec_dark_0000.fits <unknown orig name> CALPRO_DARK

Extension: IMAGE, CHIP4.INT1

Find in header: QC

find

Load Filter

Filter

Auto Display

Keyword	Value
DET.CHIP.GAIN	7.27
DET.CHIP.ID	ESO-Aladdin
DET.CHIP.LIVE	T
DET.CHIP.NO	4
DET.CHIP.NX	1024
DET.CHIP.NY	1024
DET.CHIP.PXSPACE	2.7E-05
DET.CHIP.ROX	10
DET.CHIP.X	4
DET.CHIP.XGAP	0
DET.CHIP.Y	1
DET.CHIP.YGAP	0
DET.WIN.IX	1024
DET.WIN.NY	512
DET.WIN.STARTX	1
DET.WIN.STARTY	1
QC.ROX1	6.8746721627819
QC.ROX2	6.77970798213789
QC.DARKMED	2.944444449742635

Figure 28: Master dark result within gasgano showing quality control parameters for derived readout noise.

8.2.2 Reduction of darks within esorex for flats and science frames

To obtain the same result within `esorex`, one has to have an input file containing the names of the darks, like this:

```
$ cat Darks.sof
CRIRE.2010-10-19T15:16:07.776.fits CAL_DARK
CRIRE.2010-10-19T15:16:35.681.fits CAL_DARK
CRIRE.2010-10-19T15:17:03.820.fits CAL_DARK
```

And then run `esorex` using the following command:

```
$ esorex --output-dir=/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced
        crires_spec_dark Darks.sof
```

```
***** ESO Recipe Execution Tool, version 3.9.0 *****
```

```
[ INFO ] crires_spec_dark: *****
[ INFO ] crires_spec_dark: Detector 1 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_dark: Detector 2 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_dark: Detector 3 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_dark: Detector 4 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_dark: *****
[ INFO ] crires_spec_dark: DIT value: 5 sec.
[ INFO ] crires_spec_dark: NDIT value: 3
[ INFO ] crires_spec_dark: Dark computation
[ INFO ] crires_spec_dark:   Load chip number 1

<snip>
[ INFO ] crires_spec_dark:   Collapse images for chip number 4
[ INFO ] crires_spec_dark: Division by DIT
[ INFO ] crires_spec_dark: Save the product
[ INFO ] crires_spec_dark:   Writing FITS propertylist product(CALPRO_DARK):
                                crires_spec_dark.fits
[ INFO ] esorex: Created product
                /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0000.fits
[ INFO ] esorex: 1 product created
```

Note that the output file is by default called "out_0000.fits", so within your `esorex` script you also have to have a line like the following:

```
$ mv /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0000.fits
    /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/master_dark_esorex_5p0s.fits
```

8.2.3 Inspection of pipeline results for master dark and QC parameters

Figure 29 shows the master dark created by the CRIRES pipeline. One should check that the count level looks reasonable and that the readout noise is as expected (check the CRIRES ETC

or manual). Look at the fits header information created by the pipeline in the QC (Quality Control) parameters:

```
dfits -x 0 master_dark_esorex.fits | grep QC
HIERARCH ESO QC RON1 = 3.45630348085064
HIERARCH ESO QC RON2 = 3.44496225098102
HIERARCH ESO QC DARKMED = 1.27777775128682
HIERARCH ESO QC DARKSTDEV = 0.127293764968731
HIERARCH ESO QC RON1 = 4.54955875147938
HIERARCH ESO QC RON2 = 3.53587998287295
HIERARCH ESO QC DARKMED = 2.6111110051473
HIERARCH ESO QC DARKSTDEV = 4.75389685389413
HIERARCH ESO QC RON1 = 5.70574272668766
HIERARCH ESO QC RON2 = 5.67308598788528
HIERARCH ESO QC DARKMED = 2.19444437821706
HIERARCH ESO QC DARKSTDEV = 0.240562589220506
HIERARCH ESO QC RON1 = 6.8746721627819
HIERARCH ESO QC RON2 = 6.77970798213789
HIERARCH ESO QC DARKMED = 2.94444449742635
HIERARCH ESO QC DARKSTDEV = 0.647287800699611
```

The fact that we have four readout noises, dark levels and standard deviations is due to the fact that we have four detectors. In our case the values are reasonable so we can proceed to the next step.

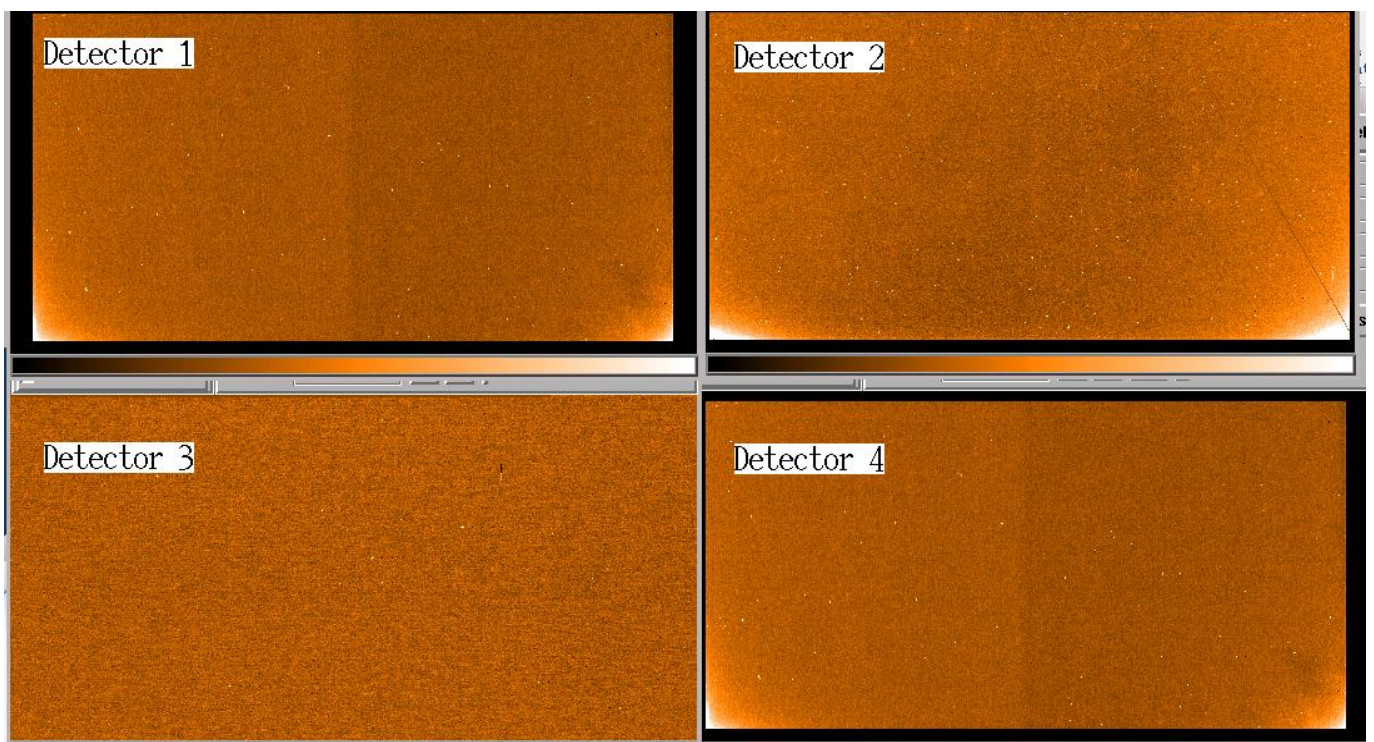


Figure 29: Master dark frame created by the ESO CRIRES pipeline. The DIT was 5-s so the glow is weak compared to the a 600-s dark image for example shown in the bottom panel of Fig. 3.

8.3 Linearity correction using `detmon_ir_lg`

The linearity correction files should be downloaded either from the archive or from here:

http://www.eso.org/observing/dfo/quality/CRIRES/pipeline/pipe_calib.html.

Note that for V2.1.x (April-2011) and later of the pipeline one must use a SINGLE non-linearity image with category COEFFS_CUBE, and for earlier versions of the pipeline (pre-April-2011 data) then you must use THREE files with categories DETLIN_A, DETLIN_B and DETLIN_C. **There are several different versions of these files depending on the date of observation**; be sure to use the one indicated on the webpage or use the provided when you run `calSelector` during the download process (see Sect. 4). In our case we need to download the following files, valid between 18th July 2010 and April 2011, viz:

```
CR_PDLA_100727A_ALL.fits (DETLIN_A)
CR_PDLB_100727A_ALL.fits (DETLIN_B)
CR_PDLC_100727A_ALL.fits (DETLIN_C)
```

These files are used by many recipes, such as `crires_spec_flat` and `crires_spec_jitter`. Section 10.4 describes the limitations of non-linearity correction for low- and high- count CRIRES observations.

If the files do not exist then you can recreate them yourself. In particular, if you have old format coefficients files that you want to use, you need to regenerate the coefficients using the `detmonp detlin` recipe on the old detector linearity raw calibration data. There used to be a `crires_util.convert` recipe in the pipeline (until 2012) to convert old coefficients files into the new ones. But it was removed. The `detmon` package is available here <http://www.eso.org/sci/software/pipelines/detmon/detmon-pipe-recipes.html>

8.4 Master flatfield creation

Typically three flatfields (classification CAL_FLAT) are taken in the day following any CRIRES science or standard observation, with the same reference wavelength. Three darks with the same DIT as the flats are also taken that should be used to create a master dark with classification CALPRO_DARK according to Sect. 8.2. Finally, the recipe accepts three non-linearity files, with classification DETLIN_A, DETLIN_B and DETLIN_C (for pre-April 2011 data) or one with classification COEFFS_CUBE (post-April 2011).

First one should visually check the (normally three) flatfields to see if the data quality looks OK, in particular that the count levels do not exceed 10,000 ADU and that there is flux on all four detectors and exposures. Some settings (particularly in the blue) suffer from vignetting on detectors 1 and 4 (Fig. 31), which is necessary to avoid contamination by adjacent orders. One should thus look at the keywords INS WLEN MIN/MAX to see where the usable range lies. Note that flatfields (especially at long wavelengths) may be affected by water vapour between the calibration lamp and the vessel entrance window. (Fig. 4) and/or ghosts (Fig. 5).

8.4.1 Reduction of flatfields within `gasgano`

Figure 31 shows the creation of flatfields using `gasgano`.

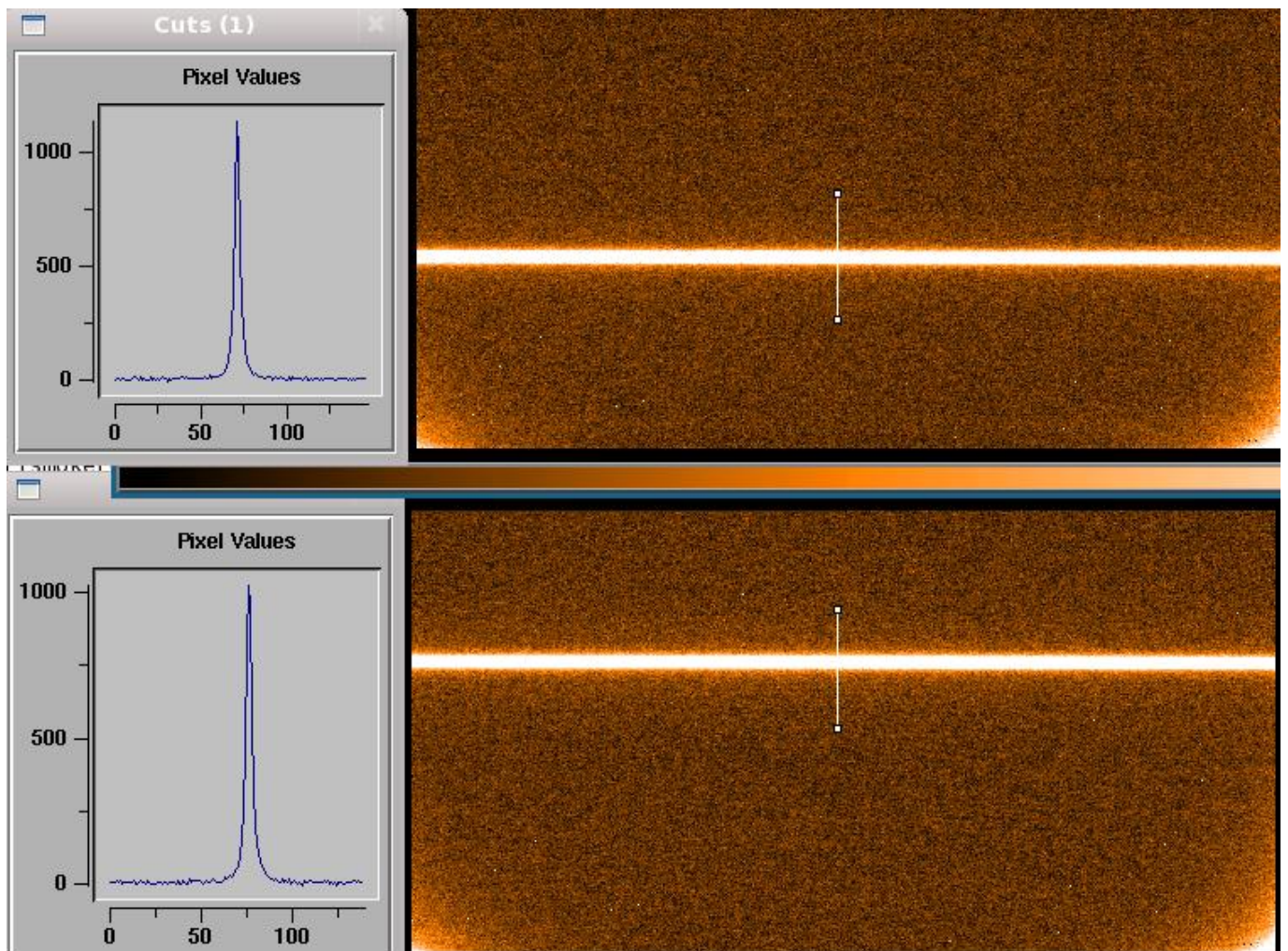


Figure 30: CRIRES nodding. The integrated flux in the two nodding positions typically should be the same as long as the atmosphere was stable between the two observations.

The screenshot shows the 'crires_spec_flat v20103' application window. It has a menu bar with 'File' and 'Help'. Below the menu bar are tabs for 'Current', 'Queued', and 'Executing'. The main interface is divided into several sections:

- Parameters:** A table with columns: Name, Value, Default, and Range.

Name	Value	Default	Range
crir. crires_spec_flat.starty	-1	-1	-1
crir. crires_spec_flat.stopy	-1	-1	-1
crir. crires_spec_flat.kappa_sigclip	3.0	3.0	3.0
crir. crires_spec_flat.lines_ratio	0.5	0.5	0.5
crir. crires_spec_flat.thresholds	0.5, 2.0	0.5, 2.0	
crir. crires_spec_flat.coil_meth	sig	sig	
crir. crires_spec_flat.replace			
crir. crires_spec_flat.clean_flat			
- Input Frames:** A table with columns: Include, Filename, Classification, Locate, and Display.

Include	Filename	Classification	Locate	Display
☒	CRIRE 2011-01-15T12:33:25.959.fits	CAL_FLAT	Locate	Display
☒	CRIRE 2011-01-15T12:34:45.559.fits	CAL_FLAT	Locate	Display
☒	CRIRE 2011-01-15T12:36:09.493.fits	CAL_FLAT	Locate	Display
☒	crires_spec_dark_0001.fits	CALPRO_DARK	Locate	Display
☒	M. CRIRES, 2010-09-21T17:04:57.320.fits	DETUN_A	Locate	Display
☒	M. CRIRES, 2010-09-21T17:05:07.420.fits	DETUN_B	Locate	Display
☒	M. CRIRES, 2010-09-21T17:05:13.214.fits	DETUN_C	Locate	Display
- Product Naming:** A section with a text field for 'Product Root Directory' (set to /home2/jsmoker/instruments/CRIRES/Pipeline/Data/Reduced), a 'Browse' button, and a 'Naming Scheme' dropdown (set to 'Numeric').
- Execute:** A large blue button with a yellow arrow icon.
- Output Frames:** A table with columns: Filename, Classification, and buttons for 'Locate' and 'Display'.

Filename	Classification	Locate	Display
crires_spec_flat_set01_0001.fits	CALPRO_FLAT		
crires_spec_flat_set01_bpm_0001.fits	CALPRO_BPM		
- Log Messages:** A scrollable text area showing the execution log.


```

01:20:19 [ INFO ] Normalise with the median
01:20:19 [ INFO ] Compute the MASTER FLAT for chip nb 4
01:20:19 [ INFO ] Correct for the dark
01:20:19 [ INFO ] Normalise with the median
01:20:19 [ INFO ] Save the products
01:20:19 [ INFO ] Writing FITS propertylist product(CALPRO_FLAT): crires_spec_flat_set01.fits
01:20:19 [ INFO ] Writing FITS propertylist product(CALPRO_BPM): crires_spec_flat_set01_bpm.fits
Product frames:
/home2/jsmoker/instruments/CRIRES/Pipeline/Data/Reduced/crires_spec_flat_set01_0001.fits
/home2/jsmoker/instruments/CRIRES/Pipeline/Data/Reduced/crires_spec_flat_set01_bpm_0001.fits
Completion status: SUCCESS
      
```

On the right side of the window, there is a 'Request Pool' section with an 'Add to pool' button and a 'Request Pool' list. Below it is an 'Execute Selected' button. A 'Clear' button is also present in the 'Output Frames' section.

Figure 31: Use of gasgano to produce a master flatfield.

8.4.2 Reduction of flatfields within esorex

For observations for pipeline versions later than April 2011, using V2.1.x or later of the pipeline then one uses the following .sof file which contains non-linearity coefficients COEFFS_CUBE, as well as the master dark with the same DIT as the flats that we prepared earlier:

```
$ cat Flatfields_1087p3_A.sof
CRIRE.2011-XX-20T10:00:53.435.fits CAL_FLAT
CRIRE.2011-XX-20T10:02:18.593.fits CAL_FLAT
CRIRE.2011-XX-20T10:03:43.485.fits CAL_FLAT
/home2/jsmoker/snip/Data/Reduced/crides_spec_dark_for_flat_0000.fits CALPRO_DARK
/home2/jsmoker/snip/Data/Reduced/detlin_coeffs_cube.fits COEFFS_CUBE
```

For observations earlier than April 2011 (as in this example), then the following .sof file should be use that contains non-linearity coefficients DETLIN_A, DETLIN_B and DETLIN_C):

```
$ cat Flatfields_1087p3_B.sof
CRIRE.2010-10-20T10:00:53.435.fits CAL_FLAT
CRIRE.2010-10-20T10:02:18.593.fits CAL_FLAT
CRIRE.2010-10-20T10:03:43.485.fits CAL_FLAT
/home2/jsmoker/snip/Data/Reduced/crides_spec_dark_0000.fits CALPRO_DARK
M.CRIRES.2010-09-21T17:04:57.320.fits DETLIN_A
M.CRIRES.2010-09-21T17:05:07.420.fits DETLIN_B
M.CRIRES.2010-09-21T17:05:13.214.fits DETLIN_C
```

And then run esorex using the following command:

```
$ esorex --output-dir=/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced
        crides_spec_flat Flatfields_1087p3_B.sof
```

```
***** ESO Recipe Execution Tool, version 3.9.0 *****
```

```
[ INFO ] crides_spec_flat: *****
[ INFO ] crides_spec_flat: Detector 1 : Y=1 => Y=512 pixels
[ INFO ] crides_spec_flat: Detector 2 : Y=1 => Y=512 pixels
[ INFO ] crides_spec_flat: Detector 3 : Y=1 => Y=512 pixels
[ INFO ] crides_spec_flat: Detector 4 : Y=1 => Y=512 pixels
[ INFO ] crides_spec_flat: *****
[ INFO ] crides_spec_flat: Reduce data set 1 / 1
[ INFO ] crides_spec_flat:   DIT value: 5 sec.
[ INFO ] crides_spec_flat:   Verify the dark DIT
[ INFO ] crides_spec_flat:   Compute the MASTER FLAT for chip nb 1
[ INFO ] crides_spec_flat:     Correct for the dark
[ INFO ] crides_spec_flat:     Normalise with the median
[ INFO ] crides_spec_flat:   Compute the MASTER FLAT for chip nb 2
[ INFO ] crides_spec_flat:     Correct for the dark
[ INFO ] crides_spec_flat:     Normalise with the median
[ INFO ] crides_spec_flat:   Compute the MASTER FLAT for chip nb 3
[ INFO ] crides_spec_flat:     Correct for the dark
```

```

[ INFO ] crires_spec_flat:      Normalise with the median
[ INFO ] crires_spec_flat:      Compute the MASTER FLAT for chip nb 4
[ INFO ] crires_spec_flat:      Correct for the dark
[ INFO ] crires_spec_flat:      Normalise with the median
[ INFO ] crires_spec_flat:      Save the products
[ INFO ] crires_spec_flat:      Writing FITS propertylist product(CALPRO_FLAT):
                                crires_spec_flat_set01.fits
[ INFO ] crires_spec_flat:      Writing FITS propertylist product(CALPRO_BPM):
                                crires_spec_flat_set01_bpm.fits
[ INFO ] esorex: Created product /home2/jsmoker/Reduced/out_0000.fits
[ INFO ] esorex: Created product /home2/jsmoker/Reduced/out_0001.fits
[ INFO ] esorex: 2 products created

$ mv /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0000.fits
  /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/master_flat_1087p3.fits
$ mv /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0001.fits
  /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/bpm_1087p3.fits

```

The first file produced is the flatfield (ideally normalised to unity, but in practice not always) with the second being a bad pixel map.

8.4.3 Inspection of pipeline results for master flatfield and QC parameters

Figure 32 shows the master flatfield produced by the ESO pipeline. In this case detectors 1 and 4 clearly are not near to unity, so the flatfielding in these chips will be very poor. One looks at the Quality Control (QC) parameters using the following dfits command

```

dfits -x 0
/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/master_flat_1087p3.fits
      | grep QC (all on one line)
HIERARCH ESO QC NBBAD = 15863
HIERARCH ESO QC FLAT MEAN = 719.548624674479
HIERARCH ESO QC FLAT STDEV = 2.00897466141822
HIERARCH ESO QC FLAT FLUX = 143.909724934896
HIERARCH ESO QC FLAT MASTER RMS = 0.485489309666914
HIERARCH ESO QC NBBAD = 3487
HIERARCH ESO QC FLAT MEAN = 4984.8427734375
HIERARCH ESO QC FLAT STDEV = 8.01957517557057
HIERARCH ESO QC FLAT FLUX = 996.9685546875
HIERARCH ESO QC FLAT MASTER RMS = 0.188135783694171
HIERARCH ESO QC NBBAD = 1053
HIERARCH ESO QC FLAT MEAN = 6063.73388671875
HIERARCH ESO QC FLAT STDEV = 7.62383087763282
HIERARCH ESO QC FLAT FLUX = 1212.74677734375
HIERARCH ESO QC FLAT MASTER RMS = 0.0843839718156275
HIERARCH ESO QC NBBAD = 86285
HIERARCH ESO QC FLAT MEAN = 2641.94441731771

```

HIERARCH ESO QC FLAT STDEV = 0.281828109551431
HIERARCH ESO QC FLAT FLUX = 528.388883463542
HIERARCH ESO QC FLAT MASTER RMS = 0.722128724207729

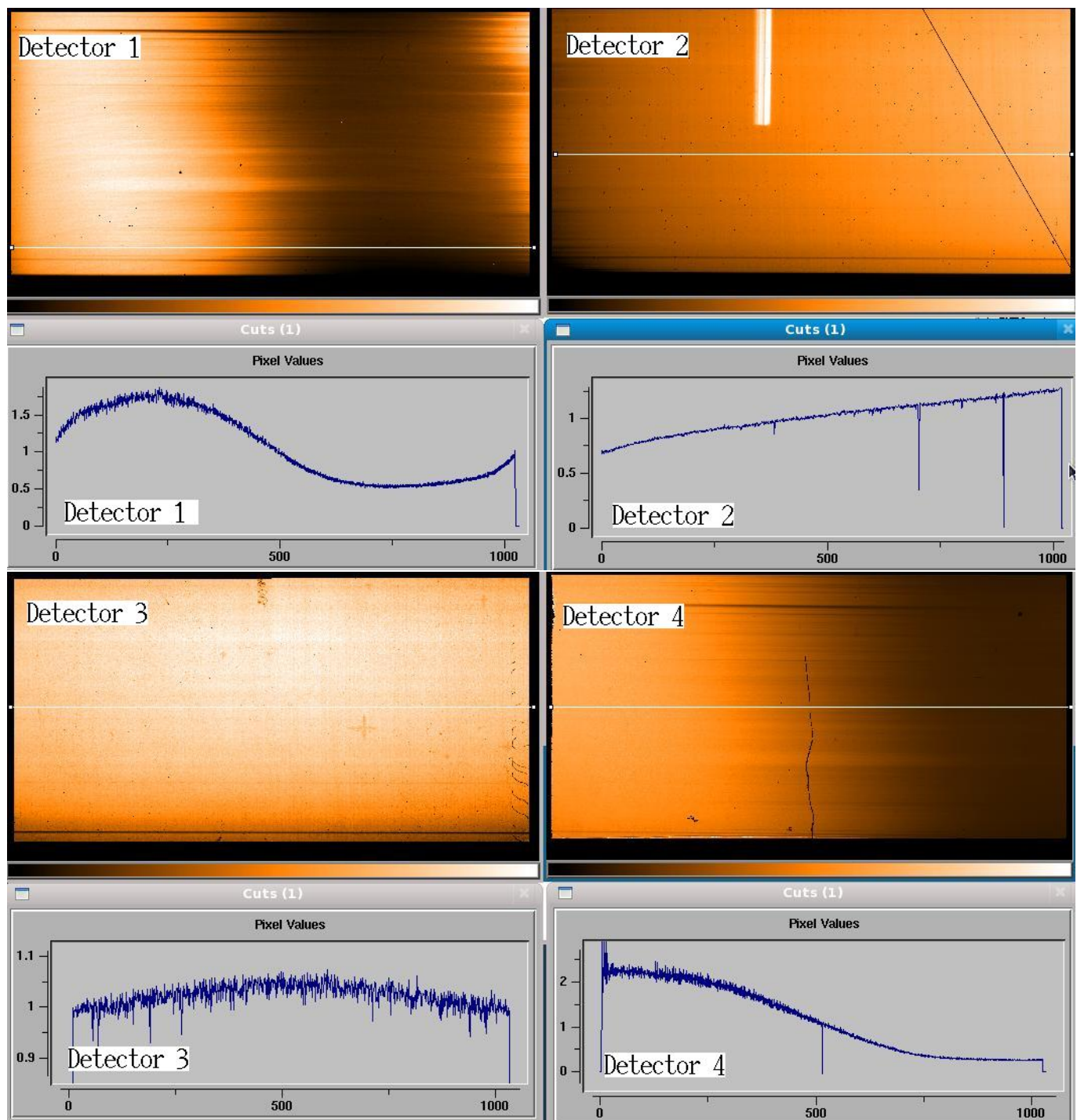


Figure 32: Master flatfield at 1087.3-nm. Detectors 1 and 4 cannot be well flatfielded as they are affected by (necessary) vignetting by the intermediate slit.

8.5 Strategies for wavelength calibration

Depending on the wavelength region used and type of wavelength calibration available, one may try different strategies.

8.5.1 Wavelength less than 2.4 microns with more than two or three well-spaced ThAr lines available per detector

This is the easiest case to calibrate well. One should use the `crires_spec_wavecal` or `crires_util_wlcalib` and `crires_util_wlassign` recipes. A good fit is shown in Fig. 33. Note that because of the limited reproducibility of the grating and prism mechanism, unless the ThAr spectrum is taken directly after the science exposure then there may be an offset between the science and ThAr wavelength calibration solutions that is impossible to verify. See Fig. 34 for an example of the typical scatter in the derived central wavelengths for two settings taken daily before the introduction of metrology.

8.5.2 Wavelength less than 2.4 microns with less than two ThAr fibres available per detector

Accurate calibration is more difficult in this case. One has several options.

- Calibrate as above (Sect. 8.5.1) using the `crires_spec_wavecal` or `crires_util_wlcalib` and `crires_util_wlassign` recipes. Because "with two points I can fit an elephant" (the two points being two ThAr features), the interpolation may lead to inaccurate results.
- Use the `crires_model_refine` recipe to improve the model. This recipe applies a detection of the Thorium/Argon emission lines from a calibration lamp exposures, and uses a cross-check of the detected lines with the lines catalogue to perform an optimisation of various model parameters (because the physical model is fit simultaneously across the spectral range of the observation it is less sensitive to the number of lines on each detector and typically requires in total only 1/4 as many lines as `crires_spec_wavecal` or `crires_util_wlcalib` to get an equivalent fit). After those parameters are optimised, they are saved in a new model configuration file that can be used to calibrate subsequent science observations.
- Use the `crires_model_fix` recipe after the initial wavelength calibration has been performed. This recipe refines the model configuration file using any CRIRES file (for its header), a TXT file with of user specified pixel position - wavelength associations and a first guess model configuration file.

8.5.3 Wavelength more than 2.4 microns with gas-cell spectra available

Because the `crires_model_refine` recipe only works with ThAr emission-line exposures (which consist of emission lines with a given (x,y) position), we have to use the `crires_model_fix` recipe. This entails interactively identifying (x,y) coordinates of known calibration features along the slit and using them in combination with a first guess model configuration file.

Filename	λ_c (nm)	O	Mean offset (pix)		Variation (pix)	
			Header	PM	Header	PM
CRIRE.2012-02-02T09:54:30.443	3456	16	25–40	15	1-5/det	1/det <5 total
CRIRE.2012-02-02T09:42:29.718	4036	14	6	12	6/det >15 total	1/det <5 total
CRIRE.2012-02-02T09:47:09.343	4539	12	7–10	0	1/det <5 total	1/det <5 total

Table 2: Residuals obtained using the CRIRES physical model and header keywords at different wavelength ranges.

8.5.4 Science frame with no gas cell or ThAr wavelength calibration and no OH or sky absorption lines

One has two possibilities here, either use the physical model or use the header keywords. Table 2 shows the results of tests to determine the absolute and relative accuracy of the physical model and header. The absolute calibration offered by the automatic header wavelength is usually not bad, however, the residual varies significantly across the detector and from detector to detector since a linear interpolation is applied to the detector start and end wavelengths. The Physical Model (PM) on the other hand typically has larger *absolute* residuals, but the *relative* calibration is good to 1-2 pixels across a detector (~ 5 pixels across the array). Hence if one knows the absolute wavelength of a stellar or interstellar feature then one can well-calibrate the whole array *a-posteriori*.

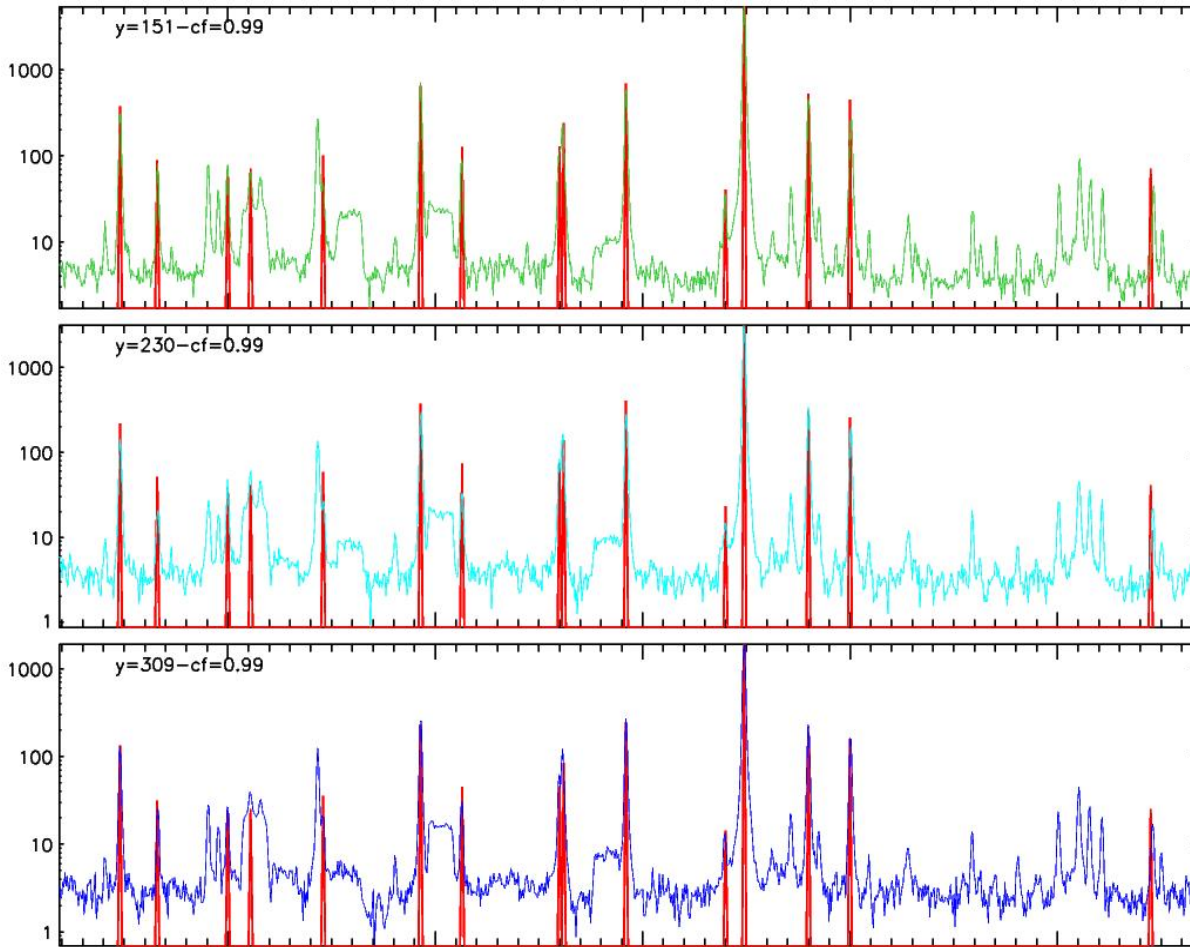


Figure 33: Well-calibrated ThAr spectrum in flux against wavelength with reference wavelength of 1111.7-nm. The figure shows the inner 3 fibres and Detector 3 only. A second order polynomial was used for the fit.

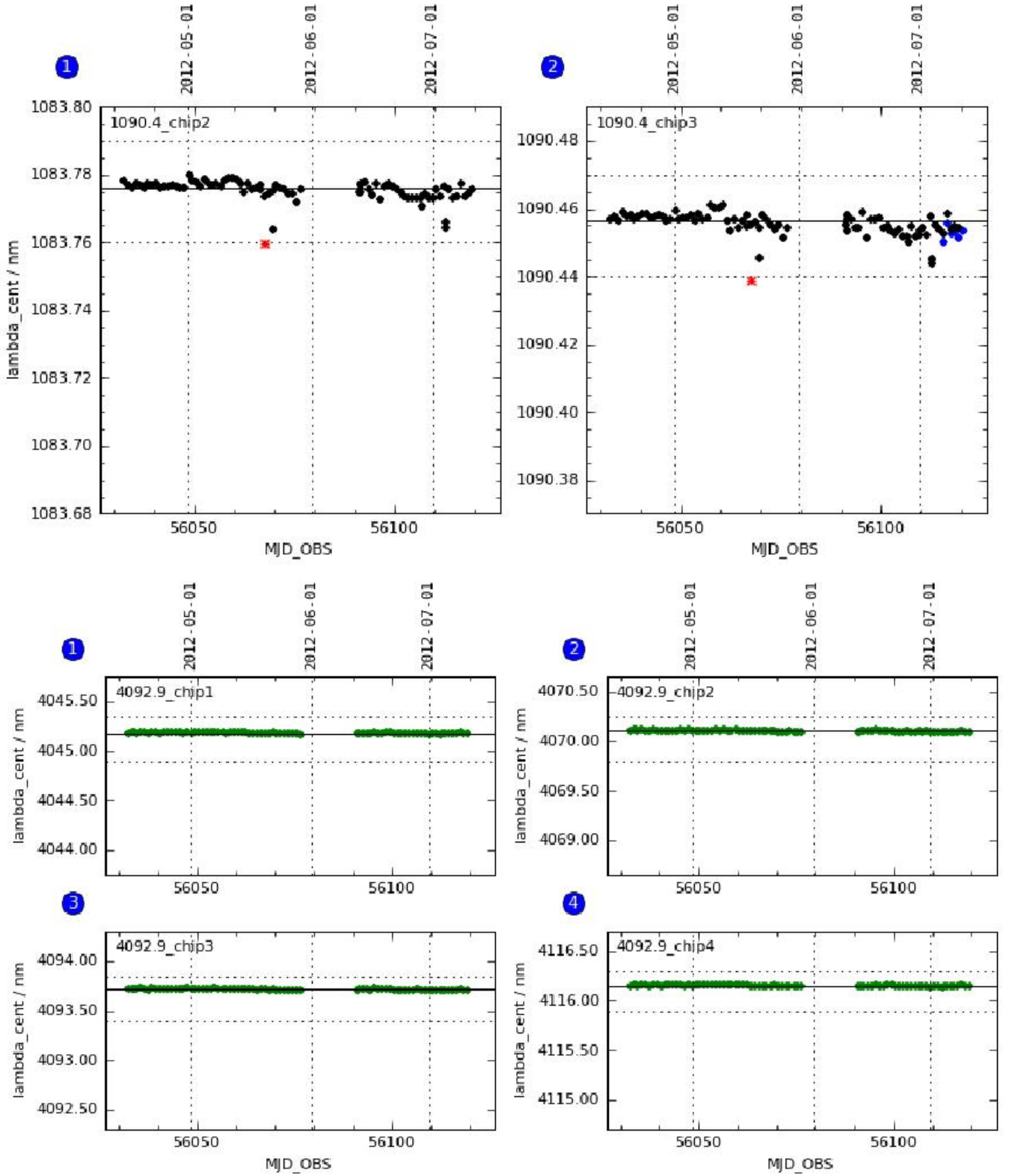


Figure 34: Central wavelength against time derived from ThAr calibrations taken daily at 1090.4-nm and N20 gas cell wavelength calibration at 4092.9-nm. The scatter on the measurements is due to the limited reproducibility of the grating and prism mechanisms.

8.6 Wavelength calibration examples

For wavelength calibration of up to 2.5 microns, ThAr spectra are typically used. Examples of such spectra are shown in Fig. 6. Wavelength calibration is one of the more difficult steps in CRIRES data reduction, and the user must be sure to **inspect the messages created by gasgano and look at the QC results created by the pipeline** in order to ensure that the results look reasonable. The top panel of Fig. 35 shows detector 3 for the 1087.3-nm ThAr calibrations. For best results you should measure by hand the 'Y' positions of the ThAr fibres (Bottom panel of Fig. 35) which are used as pipeline inputs `y_pos_c1`, `y_pos_c2`, `y_pos_c3` and `y_pos_c4` from v2.3.1 of the pipeline and starty/stopy up until v2.1.3. If you do not measure the y positions then the pipeline will use fixed values. **You must check that there are at least two or three lines on each detector, well separated in wavelength, otherwise it is likely that the wavelength calibration solution will be poor.**

8.6.1 Reduction of ThAr frames using gasgano

As a first attempt, Figure 36 shows the results of a **failed** attempt at wavelength calibration using the recipe `crires_spec_wavecal` and v2.1.3 of the pipeline. The calibration **failed** for two reasons (1) The detectors had the wrong boundaries and (2) The model configuration file was incorrect.

For the second attempt then, we (1) click the `crires_spec_wavecal.wl_nolimit` box to avoid the check of the boundaries and (2) use another model configuration file. We Execute the recipe again and although the recipe completes successfully, again if we look at the log messages we see the following messages:

```

23:13:18 [ INFO ]           Calibrate chip number 3
23:13:18 [ INFO ]           Wavelength range: 1084.7 - 1089.91
23:13:18 [ INFO ]           Correct the bad pixels
23:13:18 [ INFO ]           Correct for the dark
23:13:18 [ INFO ]           Correct for the flat field
23:13:18 [ INFO ]           Wavelength error : 60 pix = 0.305572 nm
23:13:18 [ INFO ]           Sub-Catalog extraction
23:13:18 [ INFO ]           Find the best linear solution
23:13:18 [ INFO ]           Wl error: 0.305572 nm (60 pix)
23:13:18 [ INFO ]           Nb of samples: 100
23:13:18 [ INFO ]           Nb of candidates: 10000
23:13:19 [ INFO ]           Cross-Correlation factor: 0.135488
23:13:19 [ INFO ]           Find the best degree-2 polynomial
23:13:19 [ INFO ]           Wl error: 0.0611144 nm (12 pix)
23:13:19 [ INFO ]           Nb of samples: 10
23:13:19 [ INFO ]           Nb of candidates: 1000
23:13:20 [ INFO ]           Cross-Correlation factor: 0.14317
23:13:20 [ INFO ]           Refine the degree-2 polynomial
23:13:20 [ INFO ]           Wl error: 0.0122229 nm (2.4 pix)
23:13:20 [ INFO ]           Nb of samples: 10
23:13:20 [ INFO ]           Nb of candidates: 1000
23:13:20 [ INFO ]           Cross-Correlation factor: 0.147038

```

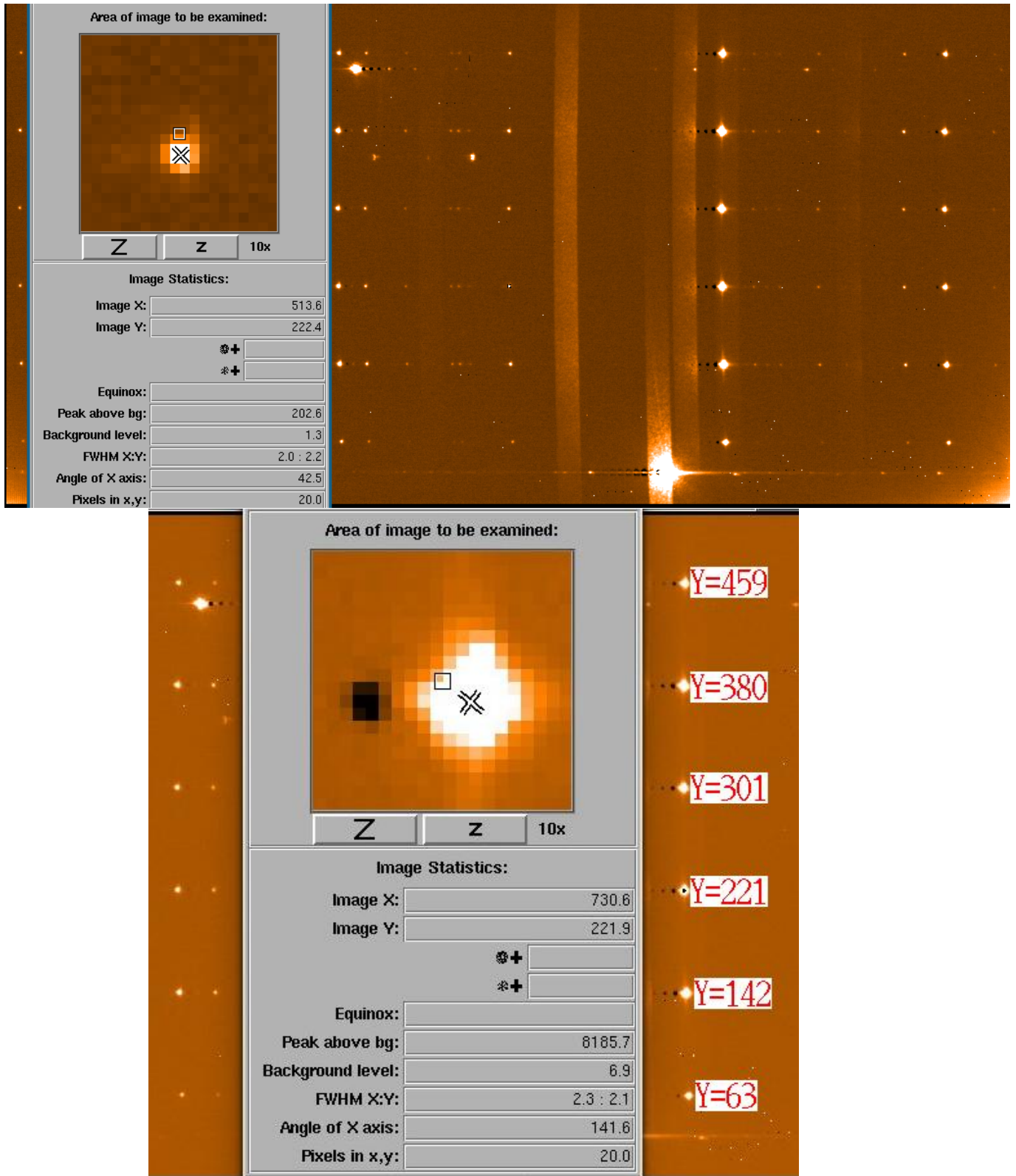


Figure 35: Top panel: CRIRES ThAr wavelength calibration for detector 3 at 1087.3-nm. The "Y" value for the ThAr lines on this detector ranges from 221 to 223 pixels. Detector 1 has 'Y' values some ~ 15 pixels higher, detector 2 values some ~ 14 pixels higher, and detector 4 values some 10 pixels lower. Bottom panel: determining the "Y" position of ThAr fibres for detector 3. From version 2.3.1 of the pipeline these are used to define the `y_pos_cN` values of the fibres in the `crires_spec.wavecal` recipe.


```

23:13:20 [ INFO ]      Refine the degree-2 polynomial
23:13:20 [ INFO ]      Wl error: 0.00244457 nm (0.48 pix)
23:13:20 [ INFO ]      Nb of samples: 10
23:13:20 [ INFO ]      Nb of candidates: 1000
23:13:20 [ INFO ]      Cross-Correlation factor: 0.147333
23:13:20 [WARNING]      Cross-correlation too low (0.147333). Use estimate

```

What can we do to improve the correlation factor? Well, instead of performing the extraction over the whole detector, we can perform the extraction of the ThAr lines in a **window**, as shown in Fig. 38 for v2.1.3 of the pipeline and 39 for versions v2.3.0/1 and upwards of the pipeline. In particular for v2.3.0 and upwards of the pipeline, if you measure the Y-position of 6 ThAr fibres on each detector then you should enter their y-coordinates in `y_pos.cN`. A wavelength solution is then obtained at each position and a 2D map of the wavelength solution is produced.

```

23:26:13 [ INFO ]      Calibrate chip number 3
23:26:13 [ INFO ]      Wavelength range: 1084.7 - 1089.91
23:26:13 [ INFO ]      Correct the bad pixels
23:26:13 [ INFO ]      Correct for the dark
23:26:13 [ INFO ]      Correct for the flat field
23:26:13 [ INFO ]      Wavelength error : 60 pix = 0.305572 nm
23:26:13 [ INFO ]      Sub-Catalog extraction
<snip>
23:26:15 [ INFO ]      Refine the degree-2 polynomial
23:26:15 [ INFO ]      Wl error: 0.0122229 nm (2.4 pix)
23:26:15 [ INFO ]      Nb of samples: 10
23:26:15 [ INFO ]      Nb of candidates: 1000
23:26:15 [ INFO ]      Cross-Correlation factor: 0.492632 (<--- Much improved)
<snip>

```

8.6.2 Reduction of ThAr frames using pipeline 2.1.3 or earlier and esorex

To reduce ThAr wavelength calibration frames using esorex we use the `crires_spec_wavecal` recipe. In order to improve the correlation coefficient (c.f. Fig. 35) we only use y-value of the data from 217 to 227 which means that the derived wavelength solution will be good only for detector 3 in this case (the position of the fibres being different on the other detectors). Note that the master dark used was taken with a 3 second DIT which corresponds to that used for the ThAr exposure.

```

$ cat ThAr_1087p3.sof
CRIRE.2010-10-20T13:59:48.752.fits      CAL_WLLAMP
CRIRE.2010-10-20T14:01:20.131.fits      CAL_WLLAMP
CRIRE.2010-10-20T14:02:47.227.fits      CAL_WLLAMP
/path/master_dark_3p0s.fits             CALPRO_DARK
/path/master_flat_1087p3.fits           CALPRO_FLAT

```

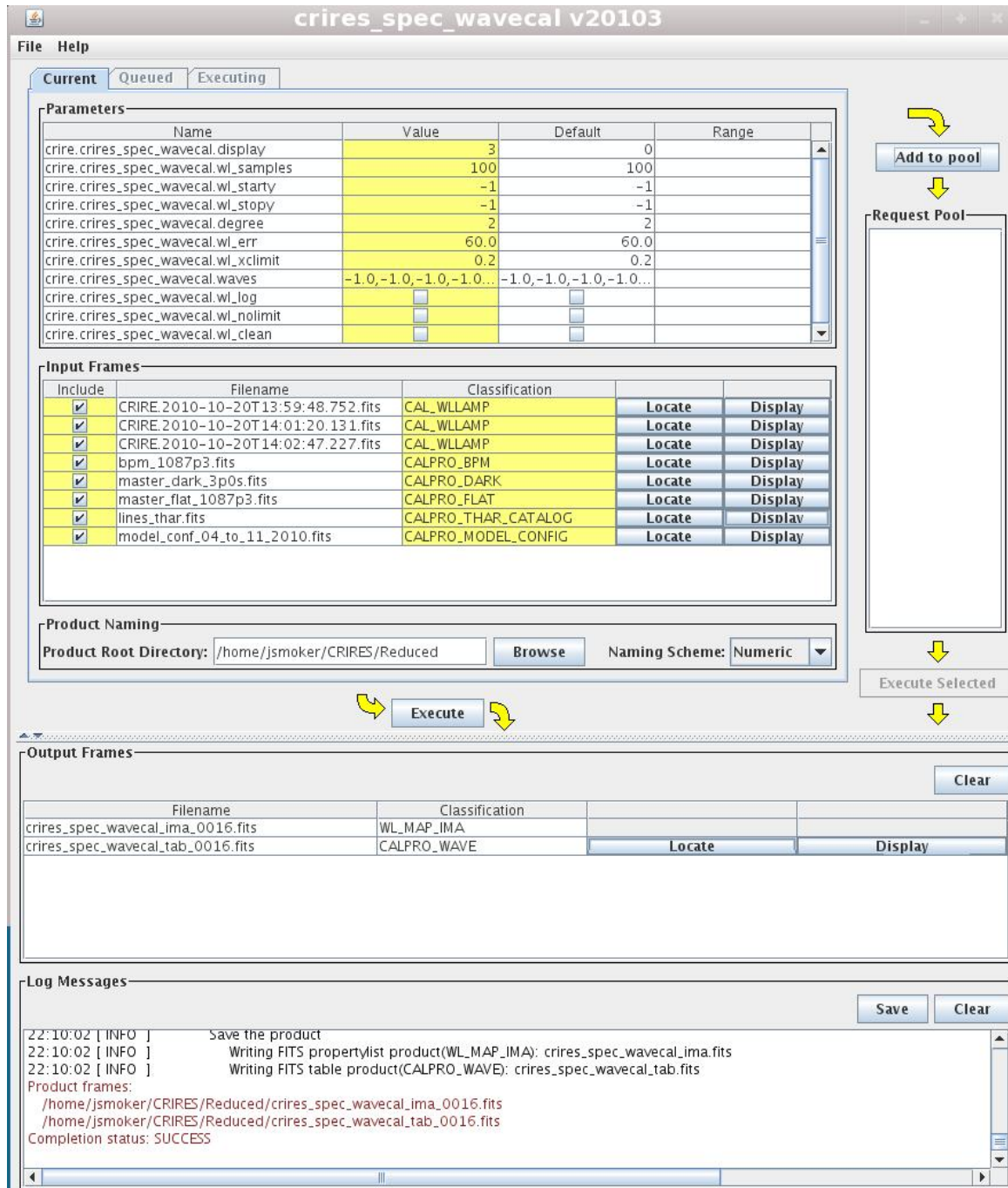


Figure 36: Apparently successful wavelength calibration using gasgano and version 2.1.3 of the pipeline, but in fact it **Failed** - see Fig. 37.

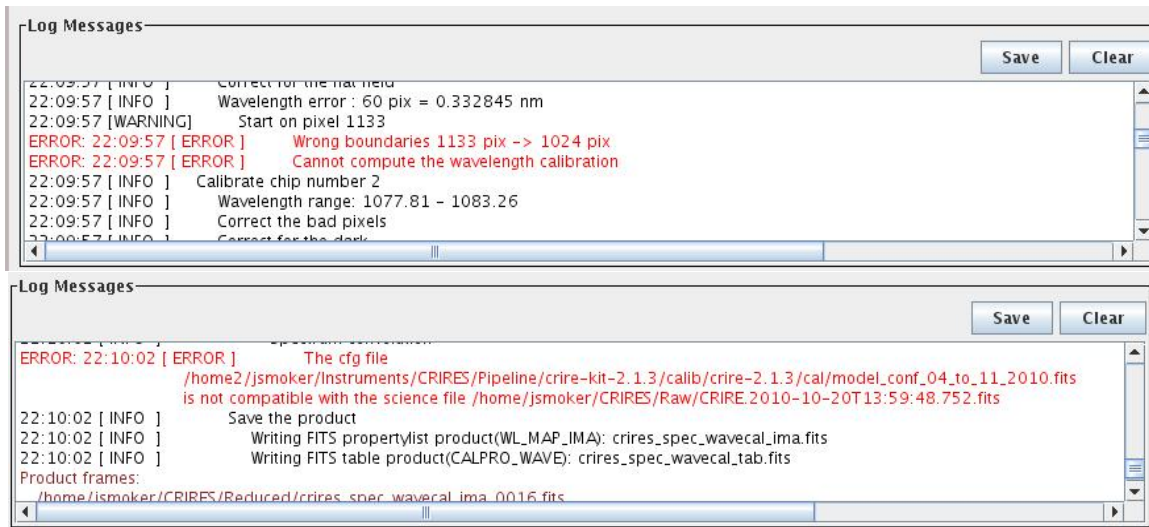


Figure 37: Inspection of log messages for wavecal reduction in Fig. 36 that show the wavelength calibration failed.

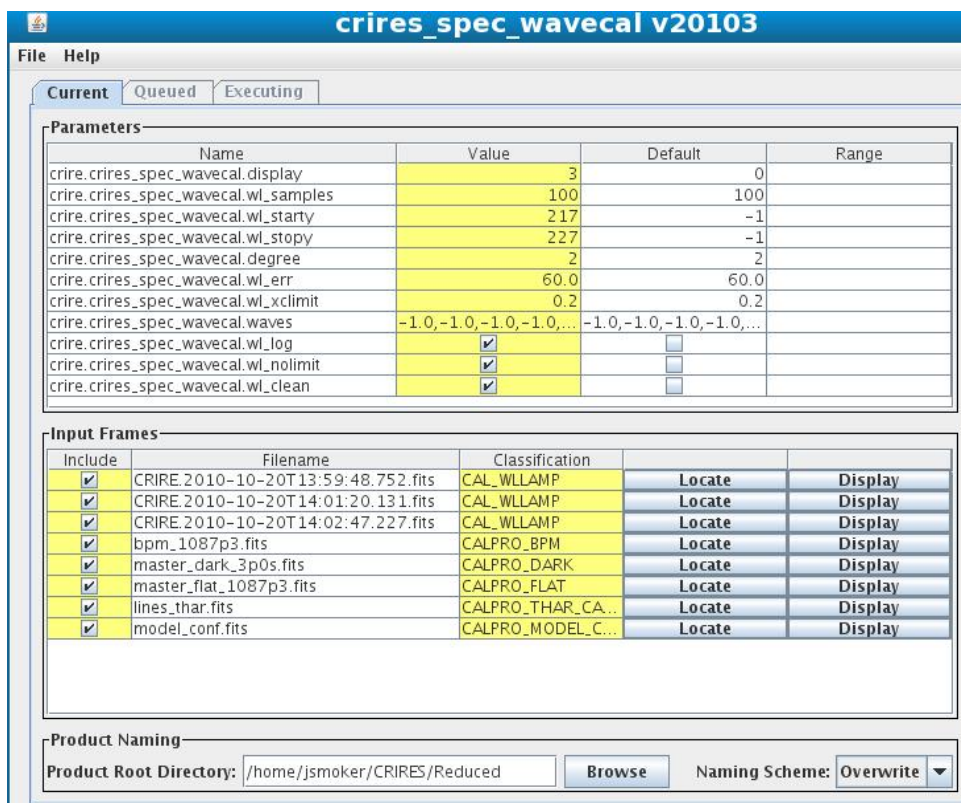


Figure 38: Successful detector 3 wavelength calibration using gasgano, changing the `crre.crires_spec_wavecal.wl_starty` and `stopx` (pipeline version 2.1.3). Detectors 1, 2 and 4 would need to be reduced separately.

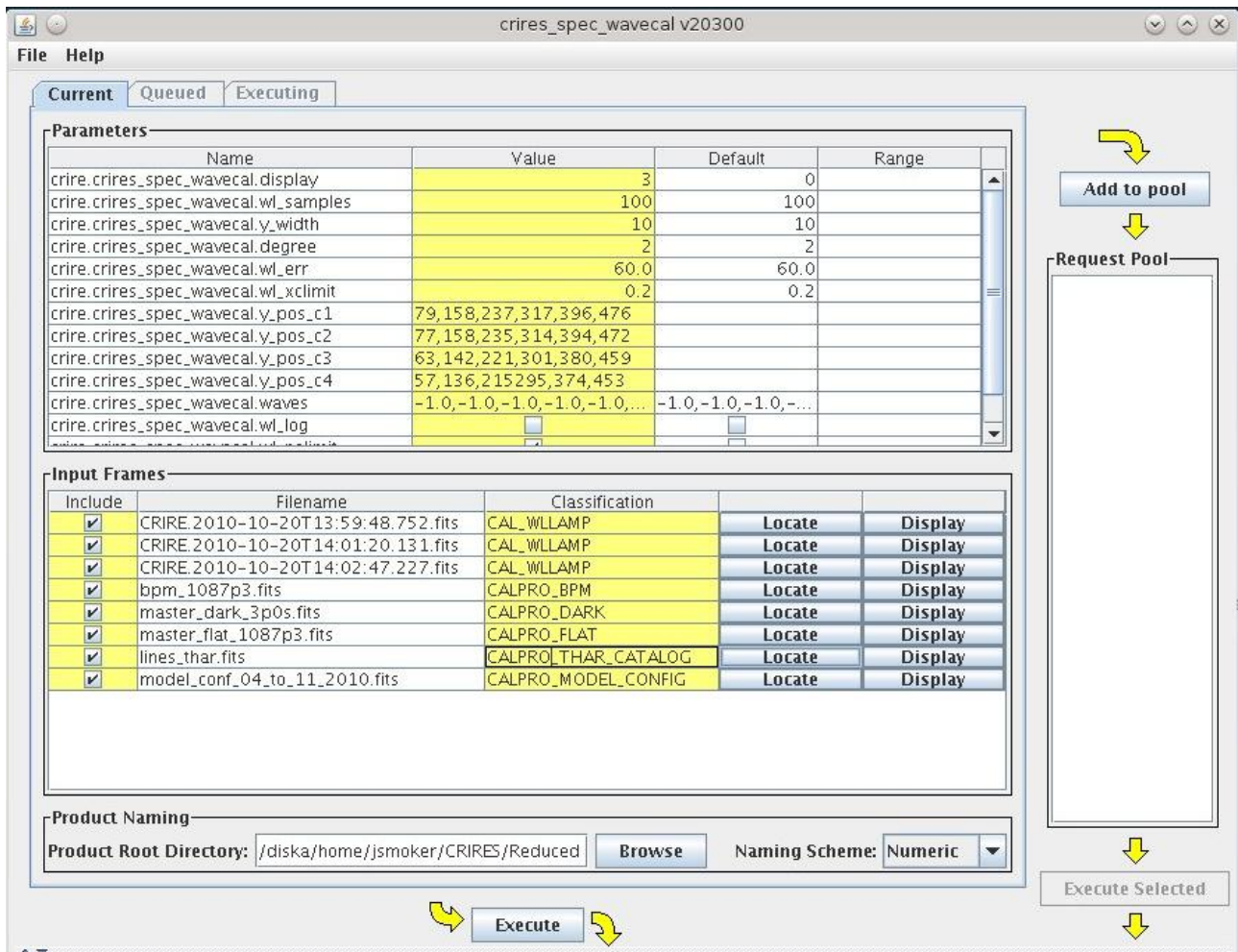


Figure 39: Wavelength calibration using gasgano using v2.3.0 of the pipeline using the measured positions of the 6 ThAr fibres and entering their y-coordinates in `y_pos_cN`. The recipe will collapse 10 rows from the input image at positions 59,158...476 on detector1, 77,158...472 on detector2, 63,142...459 on detector3 and 57,136...453 on detector 4 and all the successful calibrations are used to generate a 2D wavelength map per detector.

```

/path/bpm_1087p3.fits          CALPRO_BPM
/path/crircalib-2.1.3/cal/lines_thar.fits CALPRO_THAR_CATALOG
M.CRIRES.2010-09-21T17:04:57.320.fits    DETLIN_A
M.CRIRES.2010-09-21T17:05:07.420.fits    DETLIN_B
M.CRIRES.2010-09-21T17:05:13.214.fits    DETLIN_C

```

```

$ esorex --output-dir=/path/Reduced crires_spec_wavecals
-wl_starty=217 -wl_stop=227 -display=3 -wl_log=true -wl_nolimit=true ThAr_1087p3.sof

```

This will produce output like the following:

```

***** ESO Recipe Execution Tool, version 3.9.0 *****

[ INFO ] crires_spec_wavecals: *****
[ INFO ] crires_spec_wavecals: Detector 1 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_wavecals: Detector 2 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_wavecals: Detector 3 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_wavecals: Detector 4 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_wavecals: *****
[ INFO ] crires_spec_wavecals: Apply the Wavelength Calibration
[ INFO ] crires_spec_wavecals:   Get the calibration from the thar lamp
[ INFO ] crires_spec_wavecals:   Calibrate chip number 1
[ INFO ] crires_spec_wavecals:     Wavelength range: 1070.57 - 1076.24
[ INFO ] crires_spec_wavecals:     Correct the bad pixels
[ INFO ] crires_spec_wavecals:     Correct for the dark
[ INFO ] crires_spec_wavecals:     Correct for the flat field

<snip>
[WARNING] crires_spec_wavecals:   Cross-correlation too low (0.0640687). Use estimate
<snip>
[ INFO ] crires_spec_wavecals:   Calibrate chip number 3
[ INFO ] crires_spec_wavecals:     Wavelength range: 1084.7 - 1089.91
[ INFO ] crires_spec_wavecals:     Correct the bad pixels
[ INFO ] crires_spec_wavecals:     Correct for the dark
[ INFO ] crires_spec_wavecals:     Correct for the flat field

<snip>
[WARNING] crires_spec_wavecals:   Cannot fit a gaussian at [918, 928]
[ INFO ] crires_spec_wavecals:   Save the product
[ INFO ] crires_spec_wavecals:     Writing FITS propertylist product(WL_MAP_IMA):
                                     crires_spec_wavecals_ima.fits
[ INFO ] crires_spec_wavecals:     Writing FITS table product(CALPRO_WAVE):
                                     crires_spec_wavecals_tab.fits

[ INFO ] esorex:   Created product
                  /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0000.fits
[ INFO ] esorex:   Created product
                  /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0001.fits

```



```
[ INFO ] esorex: 2 products created
```

```
% mv /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0000.fits
    /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/ThAr_WL_MAP_IMA_1083p3.fits
% mv /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/out_0001.fits
    /home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced/ThAr_CALPRO_WAVE_1083p3.fits
```

8.6.3 Reduction of ThAr frames using pipeline 2.3.1 or later and esorex

This is the **preferred** method as it will collapse 10 rows from the input image at positions given by `y_pos_cN` and apply the cross correlation for each of the six fibre positions (e.g. around $Y=63,142,221,301,380$ and 459 in the case of detector 3). If no values of `y_pos_cN` are input then the pipeline will use fixed values; however, as the Y-position of the fibres can slightly shift it is recommended to measure them yourself and use these values in the recipe. All the successful calibrations are then used to generate a 2D wavelength map per detector.

```
$ cat ThAr_1087p3.sof
CRIRE.2010-10-20T13:59:48.752.fits      CAL_WLLAMP
CRIRE.2010-10-20T14:01:20.131.fits      CAL_WLLAMP
CRIRE.2010-10-20T14:02:47.227.fits      CAL_WLLAMP
/path/master_dark_3p0s.fits             CALPRO_DARK
/path/master_flat_1087p3.fits            CALPRO_FLAT
/path/bpm_1087p3.fits                   CALPRO_BPM
/path/crire-calib-2.1.3/cal/lines_thar.fits CALPRO_THAR_CATALOG
M.CRIRES.2010-09-21T17:04:57.320.fits    DETLIN_A
M.CRIRES.2010-09-21T17:05:07.420.fits    DETLIN_B
M.CRIRES.2010-09-21T17:05:13.214.fits    DETLIN_C

$ esorex --output-dir=/path/Reduced crires_spec_wavecal
--y_pos_c1="79,158,237,317,396,476" --y_pos_c2="77,158,235,314,396,472"
--y_pos_c3="63,142,221,301,380,459" --y_pos_c4="57,136,215,295,374,453"
--y_width=10 --display=3 --wl_log=true --wl_nolimit=true ThAr_1087p3.sof
```

8.6.4 Reduction of OH line frames using esorex

Figure 40 shows an example of a sky frame taken at a wavelength of 1111.7-nm, detector 3 only. We can use the OH lines present in the image for wavelength calibration. **The method gives variable results, due to the sparsity and weakness of OH features**, indeed the setting selected does not give a good solution so you must always check that there are enough strong lines present before proceeding. Below 1.8 microns the OH line catalogue is used, redwards of this the HITRAN model is used.

The following ".sof" file shows an example of how you would go about performing this calibration. Note that in the case that one uses a science observation with classification such as a CALNOD for OH wavelength calibration then one would (1) use y-values of the detector free

from the object spectra and (2) fool the pipeline into thinking it is a wavelength calibration by changing CAL_NOD to CAL_WLSKY in the ".sof" file or gasgano input.

```
% cat OH_Pipeline_OHcat_1111p7.sof
CRIRE.2011-07-11T22:49:11.821.fits CAL_WLSKY
/path/crIRE-kit-2.3.0/calib/crIRE-2.3.0/cal/lines_oh.fits CALPRO_OH_CATALOG
```

The following esorex command should be run in order to perform the reduction. In this case we collapse 30 lines at two positions on each of the detector, corresponding to a 10 arcsecond nodding position for the particular observations in question. One should look at the raw data (for example shown in Fig. 43) in order to work out which range of y-values to extract per detector.

```
esorex --recipe-dir=/path/crIRE-kit-2.3.0/crIRE-2.3.0/recipes/.libs
--output-dir=/path/Reduced crIRES_spec_wavecal --y_pos_c1="238,353"
--y_pos_c2="238,352" --y_pos_c3="223,337" --y_pos_c4="214,328"
--y_width=30 --display=3 OH_Pipeline_OHcat_1111p7.sof
```

The esorex command above will produce plots showing the extraction overlaid on the OH lines from the catalogue. You should compare your actual data with the catalogue (as shown in Fig. 40) to check the number of lines present. In the current case there are indeed too few lines for even a reasonable wavelength calibration.

8.6.5 Use of the physical model and esorex to improve the wavelength solution

Figure 41 shows an example of a ThAr wavelength exposure with reference wavelength of 1087.3-nm. As explained in Sect. 8.6.2, for this particular wavelength region there are sufficient lines on detector 3 for a good wavelength calibration. However, for the purposes of this explanation, let's assume that we just had **ONE** ThAr feature on detector 3, with (x,y) position of (513.6,222.4), and no other features on the other detectors. In this case we can choose to use the physical model of the instrument to obtain a wavelength solution.

The sof file needs to be like the following:

```
% cat CRIRES_model_fix_1087p3.sof
CRIRE.2010-10-20T13:59:48.752.fits MODEL_FIX_RAW
000_Model_Fix_File_ThAr_Test.txt CALPRO_MODEL_FIX_TAB
model_conf_11_2010_to_04_2011.fits CALPRO_MODEL_CONFIG
```

Note that the headers of the MODEL_FIX_RAW and CALPRO_MODEL_CONFIG are checked by the recipe to make sure that the correct model file is used. So, in the example shown the model file used is correct even though the filename of the CALPRO_MODEL_CONFIG indicates that the date is not valid.

One needs to tell the recipe the position of a line (or preferably of lines) that are measured by the user. The file needs to contain the x and y coordinates, detector number and wavelength of the line in nm.

```
% cat 000_Model_Fix_File_ThAr_Test.txt
513.6 222.4 3 1087.294870
```

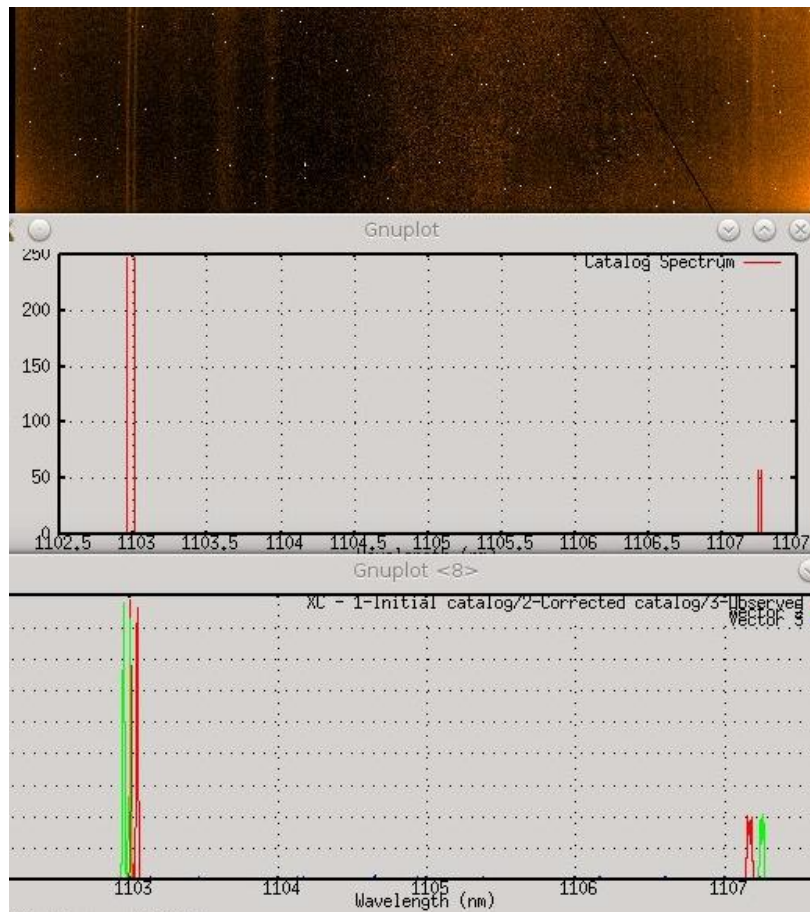



Figure 40: Example of a science frame for detector 2 with reference wavelength of 1111.7-nm.

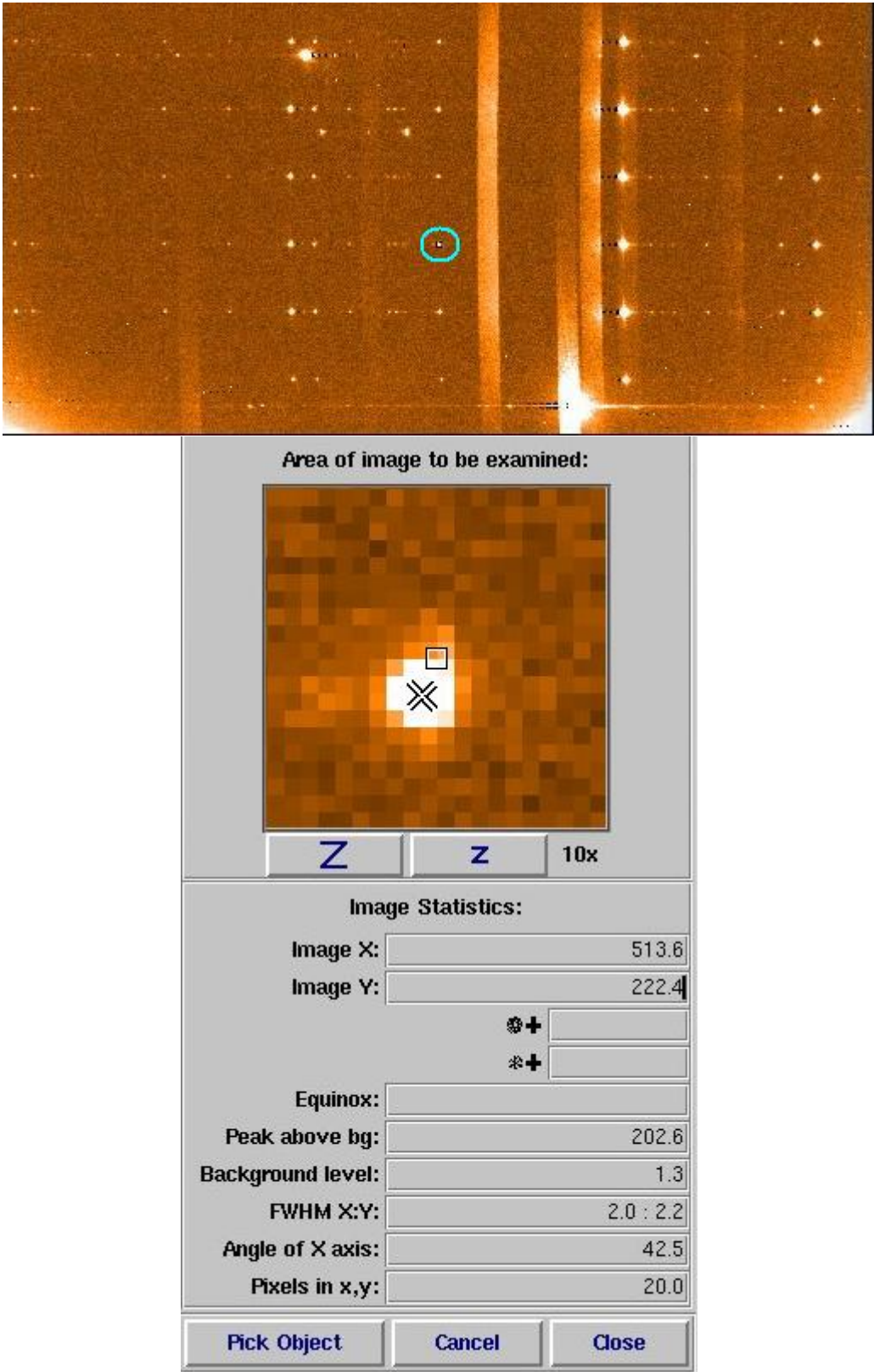


Figure 41: Example of a ThAr image for detector 3 with reference wavelength at 1087.3-nm. The cyan circle is a pick object (using the programme 'rtd') of a ThAr line at (513.6, 222.4) with wavelength 1087.294870-nm.

The recipe is run as follows:

```
esorex --recipe-dir=/path/crirc-kit-2.3.0/crirc-2.3.0/recipes/.libs
      --output-dir=/diska/home/jsmoker/CRIRES/Raw
      crircs_model_fix CRIRES_model_fix_1087p3.sof
```

...which produces output such as the following:

```
***** ESO Recipe Execution Tool, version 3.9.6 *****
[ INFO ] crircs_model_fix: [tid=000] Parse the passed TXT file
[ INFO ] crircs_model_fix: [tid=000] Compute the new configuration file
[ INFO ] crircs_model_fix: [tid=000] Get the shape of the spectra
[ INFO ] crircs_model_fix: [tid=000] slit position: -0.817091mm
check 0 1087.294870 0 -0.817091 2 2 515.193469 222.403017 513.600000 222.402359
      -1.593469 -0.000659 52 43670 191670 0.854000
[ INFO ] crircs_model_fix: [tid=000] Iterations: 66, Mean x residual: 95.756981
[ INFO ] crircs_model_fix: [tid=000] Iterations: 68, Mean x residual: 37.199280
[ INFO ] crircs_model_fix: [tid=000] Iterations: 70, Mean x residual: 16.787411
<snip>
[ INFO ] crircs_model_fix: [tid=000] Iterations: 22458, Mean x residual: 0.000001
[ INFO ] crircs_model_fix: [tid=000] Save the new configuration file
[ INFO ] crircs_model_fix: [tid=000] Writing FITS table product
      (CALPRO_MODEL_CONFIG): crircs_model_fix.fits
[ INFO ] esorex: [tid=000] Created product /diska/home/jsmoker/CRIRES/Raw/out_0000.fits
[ INFO ] esorex: [tid=000] 1 product created

% mv out_0000.fits CALPRO_MODEL_CONFIG_fixed.fits
```

We can now use this amended model file as input to the `crircs_model_wlmap` to create the 2D wavelength map or use the amended model as input to the `crircs_spec_jitter` recipe as follows:

```
% cat CRIRES_SpecJitter_1087p3_Model.sof
CRIRE.2010-10-20T03:21:52.798.fits      CAL_NOD
CRIRE.2010-10-20T03:22:23.810.fits      CAL_NOD
/path/Reduced/bpm_1087p3.fits           CALPRO_BPM
/path/Reduced/master_dark_esorex_5p0s.fits CALPRO_DARK
/path/Reduced/master_flat_1087p3.fits    CALPRO_FLAT
CALPRO_MODEL_CONFIG_fixed.fits          CALPRO_MODEL_CONFIG
M.CRIRES.2010-09-21T17:04:57.320.fits    DETLIN_A
M.CRIRES.2010-09-21T17:05:07.420.fits    DETLIN_B
M.CRIRES.2010-09-21T17:05:13.214.fits    DETLIN_C
/path/crirc-2.1.3/cal/stdstars.fits      CALPRO_STD_PHOTOFLUX

% esorex --recipe-dir=/Path/crirc-2.3.0/recipes/.libs --output-dir=/path/CRIRES/Raw
      crircs_spec_jitter -display=3 CRIRES_SpecJitter_1087p3_Model.sof
```

```
***** ESO Recipe Execution Tool, version 3.9.6 *****
```

```

[ INFO ] crires_spec_jitter: [tid=000] *****
[ INFO ] crires_spec_jitter: [tid=000] Detector 1 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: [tid=000] Detector 2 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: [tid=000] Detector 3 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: [tid=000] Detector 4 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: [tid=000] *****
[ INFO ] crires_spec_jitter: [tid=000] Images combination
<snip>

[ ERROR ] crires_spec_jitter: [tid=000] The catalogs are missing
[ INFO ] crires_spec_jitter: [tid=000] No computed solution available - uniform Map
[ INFO ] crires_spec_jitter: [tid=000] No computed solution available - uniform Map
[ INFO ] crires_spec_jitter: [tid=000] No computed solution available - uniform Map
[ INFO ] crires_spec_jitter: [tid=000] No computed solution available - uniform Map
[ ERROR ] crires_spec_jitter: [tid=000] Missing CFG-IDENT in CALPRO_MODEL_CONFIG_fixed.fits
[ INFO ] crires_spec_jitter: [tid=000] Call the model to get the wavelengths

[ INFO ] crires_spec_jitter: [tid=000] slit position: -0.446678mm
[ INFO ] crires_spec_jitter: [tid=000] Sensitivity / Conversion / Throughput computation
[ INFO ] crires_spec_jitter: [tid=000] Search a star around RA=37.0396,
                                DEC=8.46107 in the catalog
<snip>
[ INFO ] esorex: [tid=000] Created product /diska/home/jsmoker/CRIRES/Raw/out_0004.fits
[ INFO ] esorex: [tid=000] Created product /diska/home/jsmoker/CRIRES/Raw/out_0005.fits
[ INFO ] esorex: [tid=000] 6 products created

% mv out_0001.fits SpecJit_1087p4_PipeDefaults_STD_COMBINED_IMA_Model.fits
% mv out_0002.fits SpecJit_1087p4_PipeDefaults_STD_CONTRIBUTION_IMA_Model.fits
% mv out_0003.fits SpecJit_1087p4_PipeDefaults_STD_PROFILE_IMA_Model.fits
% mv out_0004.fits SpecJit_1087p4_PipeDefaults_STD_WL_MAP_IMA_Model.fits
% mv out_0005.fits SpecJit_1087p4_PipeDefaults_STD_EXTRACT_SENS_TAB_Model.fits

```

8.6.6 Inspection of the pipeline results produced by crires_spec_wave to check on the wavelength calibration solution

As written in the CRIRES Pipeline Users Manual, the recipe produces a 4 extensions file with tables containing the polynomial coefficients giving the relation $\text{wavelength} = F(\text{pixel})$ (CALPRO_WAVE). It also produces 2 wavelength maps, the one obtained by cross correlation (WL_MAP_IMA) and the one computed by the model (WL_MAP_MODEL_IMA). Figure 42 shows plots produced by the pipeline of the extraction for detector 3. One should check that:

- The detected lines correspond to actual ThAr fibres on the spectra.
- The computed initial solution difference wavelength does not show large values; typically they should be $< 0.1\text{-nm}$.
- The initial, computed and solution fits are similar.

One should also look at the headers to check the cross-correlation. Small numbers (< 0.5) mean that likely the solution is bad. In this case the Detector 3 cross-correlation result was 0.83 so the fit is good and wavelength solution can be trusted.

```
$ dfits -x 0 crires_spec_wavecal_ima.fits | grep XCORR
HIERARCH ESO QC XCORR =      -1.
HIERARCH ESO QC XCORR =      -1.
HIERARCH ESO QC XCORR = 0.834059022839282
HIERARCH ESO QC XCORR = 0.208174330229398

$ dfits -x 0 crires_spec_wavecal_ima.fits | grep RESOL
HIERARCH ESO QC RESOL MED = 239922.774365529
HIERARCH ESO QC RESOL MED = 140029.604526438
HIERARCH ESO QC RESOL MED = 221013.021706264
HIERARCH ESO QC RESOL MED = 188169.29772676
```

8.7 Combining nodding spectra

We combine nodding spectra such as Fig. 43 using the recipe `crires_spec_jitter`. As input this needs a set of nodding positions, plus optionally the bad pixel map, master dark and flatfield, wavelength solution and detector linearity coefficients. In the case of a standard star observation, the fits file containing the flux against wavelength may also be included.

One can also perform the extraction of the individual nodding positions in order to retain full resolution, and also it is possible to run the recipe on only one spectrum in OBS_DIR mode (see 7.3). **Note that from version 2.3.1 of the pipeline (preferred), the wavelength calibration is applied at the different Y positions and ThAr pinhole data are fully supported.**

First, have a look at the raw spectra and check that:

- The counts do not exceed 10,000 ADU on any of the four detectors (including sky). If they do exceed this quantity the non-linearity correction will not be so good and pipeline results should be treated with caution.
- The counts in the two nods are similar.

Figure 43 shows the counts in two nodding positions for detector 3 only.

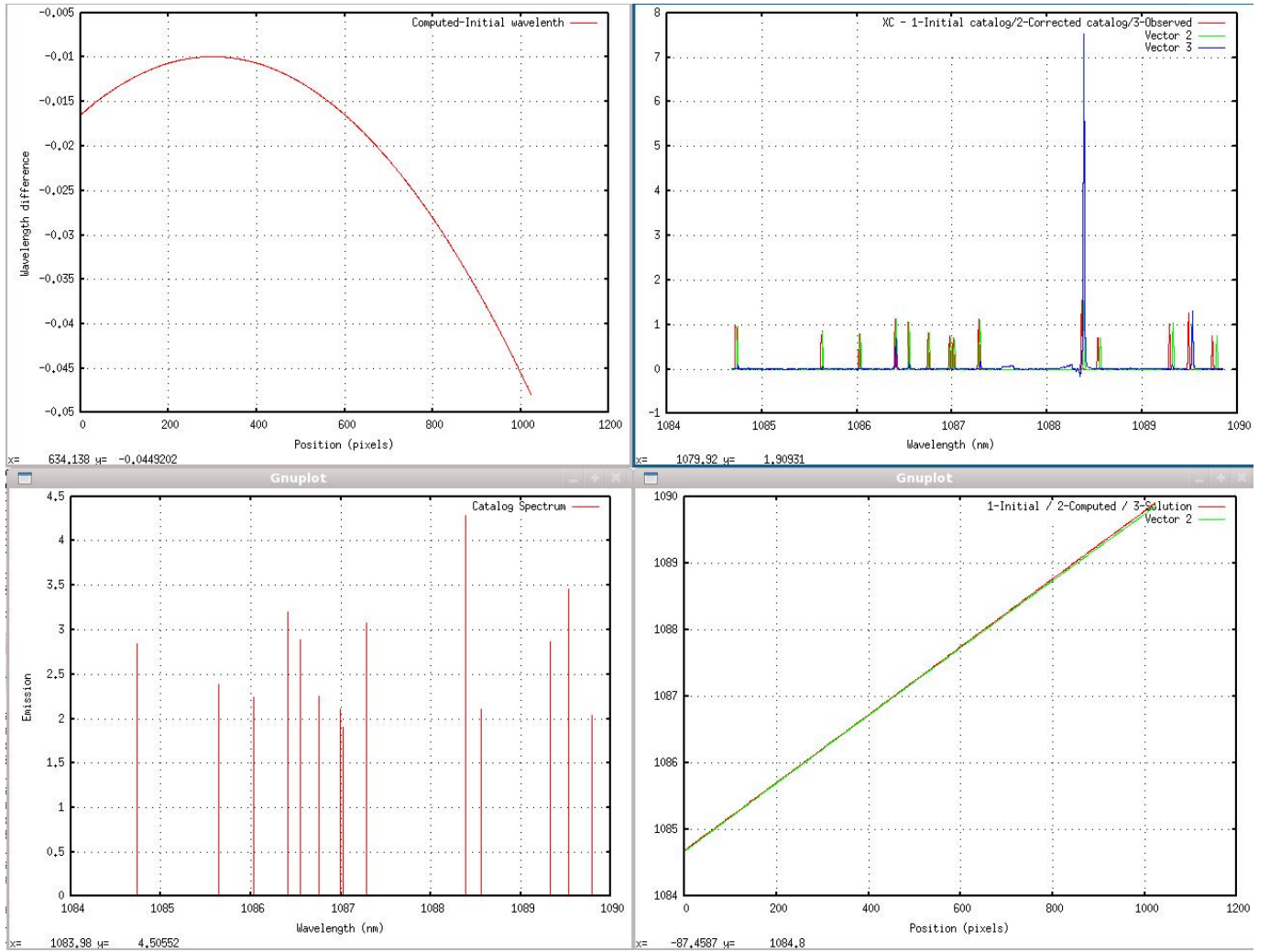


Figure 42: Results of the CRIRES pipeline wavelength calibration fit for Detector 3, 1087.3-nm. These plots are produced directly from the pipeline using the "display=N" option, with N being the detector number 1-4.

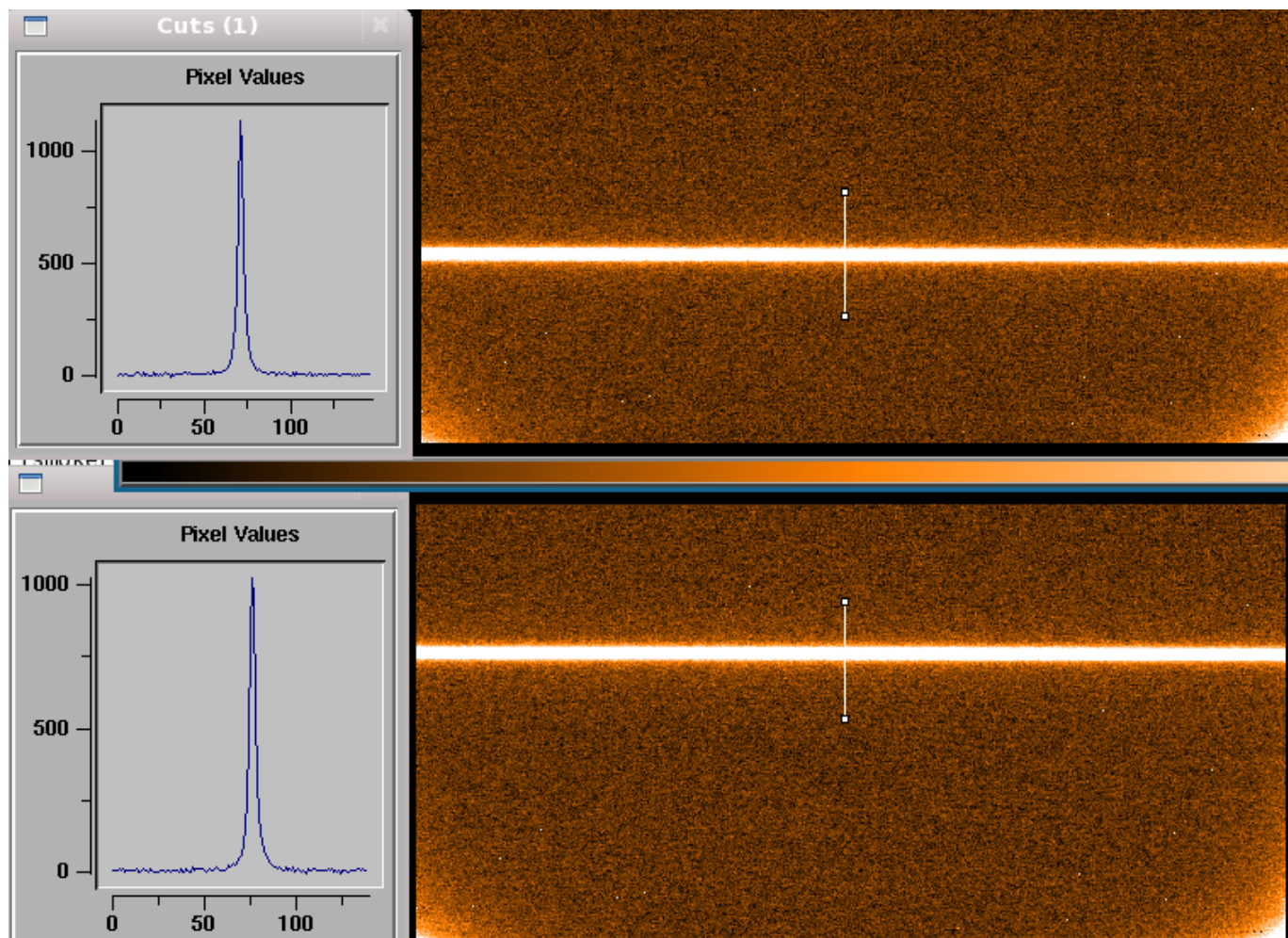


Figure 43: Figure showing detector 3 only in the two nodding positions for 1087.3-nm observations. The counts in the two nodding positions are similar. Note that the Y-position of the traces on the four detectors is different.

8.7.1 Reduction of nodding frames using gasgano

Figure 44 shows the reduction of a single AB pair using the `crires_spec_jitter` recipe within `gasgano`. In the example shown we just use the default values which produce a reduced spectrum.

8.7.2 Reduction of nodding frames using esorex

For a basic reduction one needs a sof file that looks like the following (with the individual entries on one line each, here linebreaks included for clarity):

```
$ cat CRIRES_SpecJitter_1087p3.sof
CRIRES.2010-10-20T03:21:52.798.fits          CAL_NOD
CRIRES.2010-10-20T03:22:23.810.fits          CAL_NOD
/home2/jsmoker/Instruments/CRIRES/Pipeline/Data
                                         /Reduced/bpm_1087p3.fits          CALPRO_BPM
/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced
                                         /master_dark_esorex_5p0s.fits  CALPRO_DARK
/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced
                                         /master_flat_1087p3.fits      CALPRO_FLAT
/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced
                                         /ThAr_CALPRO_WAVE_1083p3.fits CALPRO_WAVE
M.CRIRES.2010-09-21T17:04:57.320.fits          DETLIN_A
M.CRIRES.2010-09-21T17:05:07.420.fits          DETLIN_B
M.CRIRES.2010-09-21T17:05:13.214.fits          DETLIN_C
/home2/jsmoker/Instruments/CRIRES/Pipeline/crre-kit-2.1.3
                                         /calib/crre-2.1.3/cal/stdstars.fits  CALPRO_STD_PHOTOFLUX
```

And then run the `esorex` using the following command:

```
$ esorex --output-dir=/home2/jsmoker/Instruments/CRIRES/Pipeline/Data/Reduced
crires_spec_jitter -display=3 CRIRES_SpecJitter_1087p3.sof
```

```
***** ESO Recipe Execution Tool, version 3.9.0 *****
```

```
[ INFO ] crires_spec_jitter: *****
[ INFO ] crires_spec_jitter: Detector 1 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: Detector 2 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: Detector 3 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: Detector 4 : Y=1 => Y=512 pixels
[ INFO ] crires_spec_jitter: *****
[ INFO ] crires_spec_jitter: Images combination
[ INFO ] crires_spec_jitter: Load the chip number 1
[ INFO ] crires_spec_jitter: Correct the bad pixels
[ INFO ] crires_spec_jitter: Correct for the dark
[ INFO ] crires_spec_jitter: Correct for the flat field
```

<snip>

```

[ INFO ] crires_spec_jitter: Throughput computation
[ INFO ] crires_spec_jitter: Exposure time: 20 sec.
[ INFO ] crires_spec_jitter: Sensitivity computation
[ INFO ] crires_spec_jitter: Save the product
[ INFO ] crires_spec_jitter: Writing FITS propertylist product(STD_COMBINED_IMA):
                             crires_spec_jitter_comb.fitsspectrums
[ INFO ] crires_spec_jitter: Writing FITS propertylist product(STD_CONTRIBUTION_IMA):
                             crires_spec_jitter_contrib.fits
[ INFO ] crires_spec_jitter: Writing FITS propertylist product(STD_PROFILE_IMA):
                             crires_spec_jitter_prof.fits
[ INFO ] crires_spec_jitter: Writing FITS propertylist product(STD_BGD_MAP_IMA):
                             crires_spec_jitter_bgmap.fits
[ INFO ] crires_spec_jitter: Writing FITS propertylist product(STD_WL_MAP_IMA):
                             crires_spec_jitter_wlmap.fits
[ INFO ] crires_spec_jitter: Writing FITS table product(STD_EXTRACT_SENS_TAB):
                             crires_spec_jitter_extracted.fits
[ INFO ] esorex: Created product /home2/jsmoker/snip/Data/Reduced/out_0000.fits
[ INFO ] esorex: Created product /home2/jsmoker/snip/Data/Reduced/out_0001.fits
[ INFO ] esorex: Created product /home2/jsmoker/snip/Data/Reduced/out_0002.fits
[ INFO ] esorex: Created product /home2/jsmoker/snip/Data/Reduced/out_0003.fits
[ INFO ] esorex: Created product /home2/jsmoker/snip/Data/Reduced/out_0004.fits
[ INFO ] esorex: Created product /home2/jsmoker/snip/Data/Reduced/out_0005.fits

$ mv /home2/jsmoker/snip/Data/Reduced/out_0000.fits
    /home2/jsmoker/snip/Data/Reduced/SpecJit_1087p4_PipeDefaults_STD_COMBINED_IMA.fits
$ mv /home2/jsmoker/snip/Reduced/out_0001.fits
    /home2/jsmoker/snip/Reduced/SpecJit_1087p4_PipeDefaults_STD_CONTRIBUTION_IMA.fits
$ mv /home2/jsmoker/snip/Reduced/out_0002.fits
    /home2/jsmoker/snip/Reduced/SpecJit_1087p4_PipeDefaults_STD_PROFILE_IMA.fits
$ mv /home2/jsmoker/snip/Reduced/out_0003.fits
    /home2/jsmoker/snip/Reduced/SpecJit_1087p4_PipeDefaults_STD_BGD_MAP_IMA.fits
$ mv /home2/jsmoker/snip/Reduced/out_0004.fits
    /home2/jsmoker/snip/Reduced/SpecJit_1087p4_PipeDefaults_STD_WL_MAP_IMA.fits
$ mv /home2/jsmoker/snip/Reduced/out_0005.fits
    /home2/jsmoker/snip/Reduced/SpecJit_1087p4_PipeDefaults_STD_EXTRACT_SENS_TAB.fits

```

8.7.3 Inspection of pipeline results produced by crires_spec_jitter

The results for such an extraction are shown in Fig. 45, with the same reduction without any calibration applied also being displayed. Note that these spectra are the combination of the two nodding positions. Figure 46 shows results of optimal and rectangular extraction. Finally, Fig. 47 shows the results of the extraction using `crires_spec_jitter.comb_used = "NODA", "NODB"` or `"COMB"`. Note that this figure (and the wavelength calibration) was created using version 2.1.3 of the pipeline that did not apply the different wavelength solutions at different 'Y' positions on the detectors. Versions 2.3.1 and above take this correction into account.

crires_spec_jitter v20103

File Help

Current Queued Executing

Parameters

Name	Value	Default	Range
crire.crires_spec_jitter.display	0	0	
crire.crires_spec_jitter.hor_size	31	31	
crire.crires_spec_jitter.spec_hsize	20	20	
crire.crires_spec_jitter.closing_hs	2	2	
crire.crires_spec_jitter.wl_samples	100	100	
crire.crires_spec_jitter.wl_starty	-1	-1	
crire.crires_spec_jitter.wl_stopy	-1	-1	
crire.crires_spec_jitter.degree	2	2	
crire.crires_spec_jitter.kappa	1.0	1.0	
crire.crires_spec_jitter.clean_rate	0.5	0.5	
crire.crires_spec_jitter.wl_err	60.0	60.0	
crire.crires_spec_jitter.wl_xclimit	0.2	0.2	
crire.crires_spec_jitter.waves	-1.0,-1.0,-1.0,-1.0,-1.0,-...	-1.0,-1.0,-1.0,-1.0,-1.0,-...	
crire.crires_spec_jitter.rej	0,0	0,0	
crire.crires_spec_jitter.spec_zone	-1,-1	-1,-1	
crire.crires_spec_jitter.comb_used	COMB	COMB	
crire.crires_spec_jitter.refine	☒	☒	
crire.crires_spec_jitter.onlyA	☐	☐	
crire.crires_spec_jitter.onlyB	☐	☐	
crire.crires_spec_jitter.blind	☐	☐	
crire.crires_spec_jitter.wl_log	☐	☐	
crire.crires_spec_jitter.wl_nolimit	☐	☐	
crire.crires_spec_jitter.wl_clean	☐	☐	

Input Frames

Include	Filename	Classification		
☒	CRIRE.2010-10-20T03:21:52.798.fits	CAL_NOD	Locate	Display
☒	CRIRE.2010-10-20T03:22:23.810.fits	CAL_NOD	Locate	Display
☒	bpm_1087p3.fits	CALPRO_BPM	Locate	Display
☒	master_dark_5p0s.fits	CALPRO_DARK	Locate	Display
☒	master_flat_1087p3.fits	CALPRO_FLAT	Locate	Display
☒	crires_spec_wavecal_tab_0022.fits	CALPRO_WAVE	Locate	Display
☒	M.CRILES.2010-09-21T17:04:57.320.fits	DETLIN_A	Locate	Display
☒	M.CRILES.2010-09-21T17:05:07.420.fits	DETLIN_B	Locate	Display
☒	M.CRILES.2010-09-21T17:05:13.214.fits	DETLIN_C	Locate	Display

Product Naming

Product Root Directory: Browse Naming Scheme:

Execute

Figure 44: Dark subtraction, flatfielding, wavelength calibration, non-linearity correction, extraction and combination of simple nodding observations using gasgano.

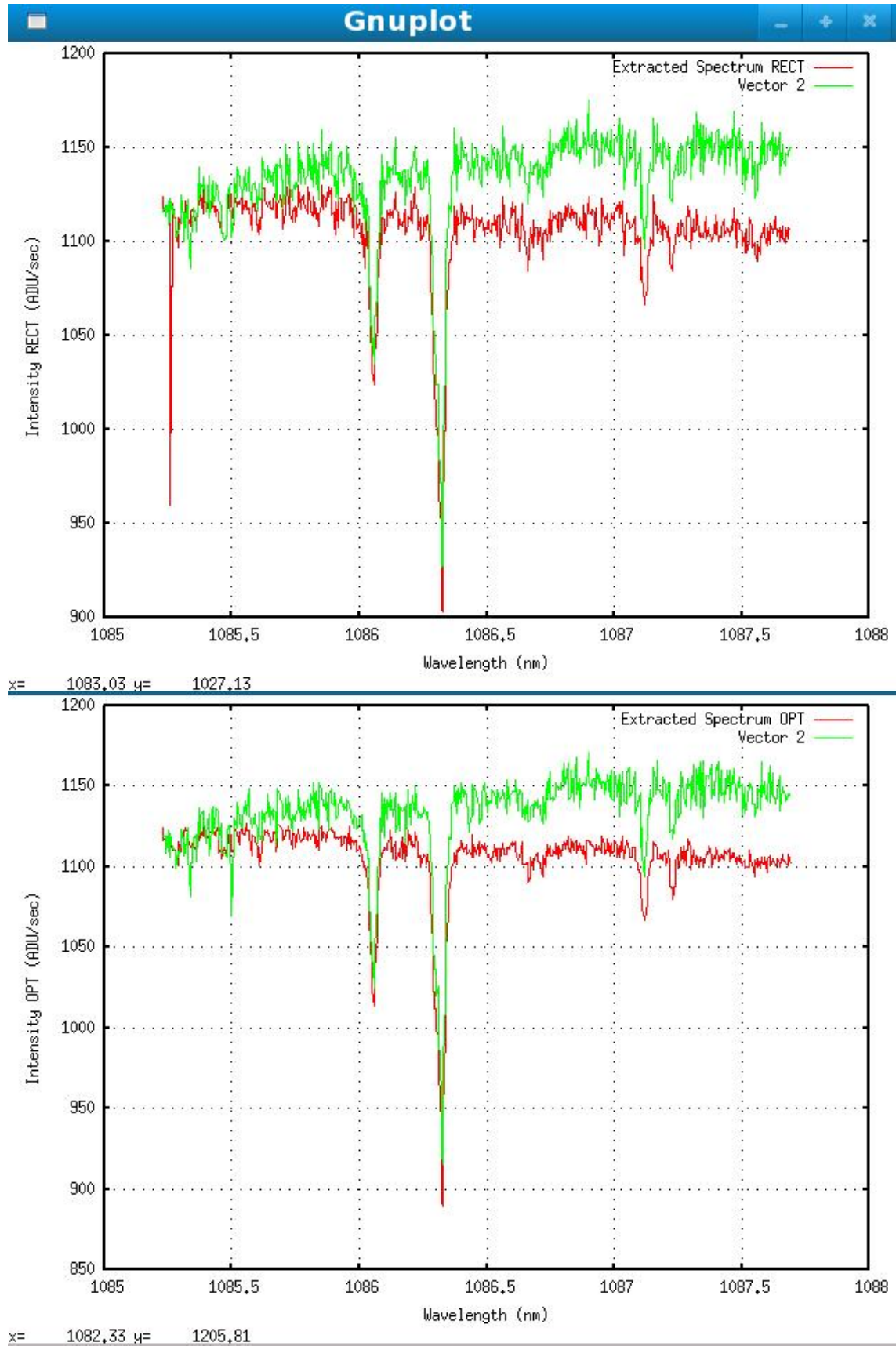


Figure 45: Comparison of output from `crires_spec_jitter` for uncalibrated observations (green), and after full reduction (red), both for rectangular and optimal extraction (detector 3 only). The improvement in S/N ratio is apparent.

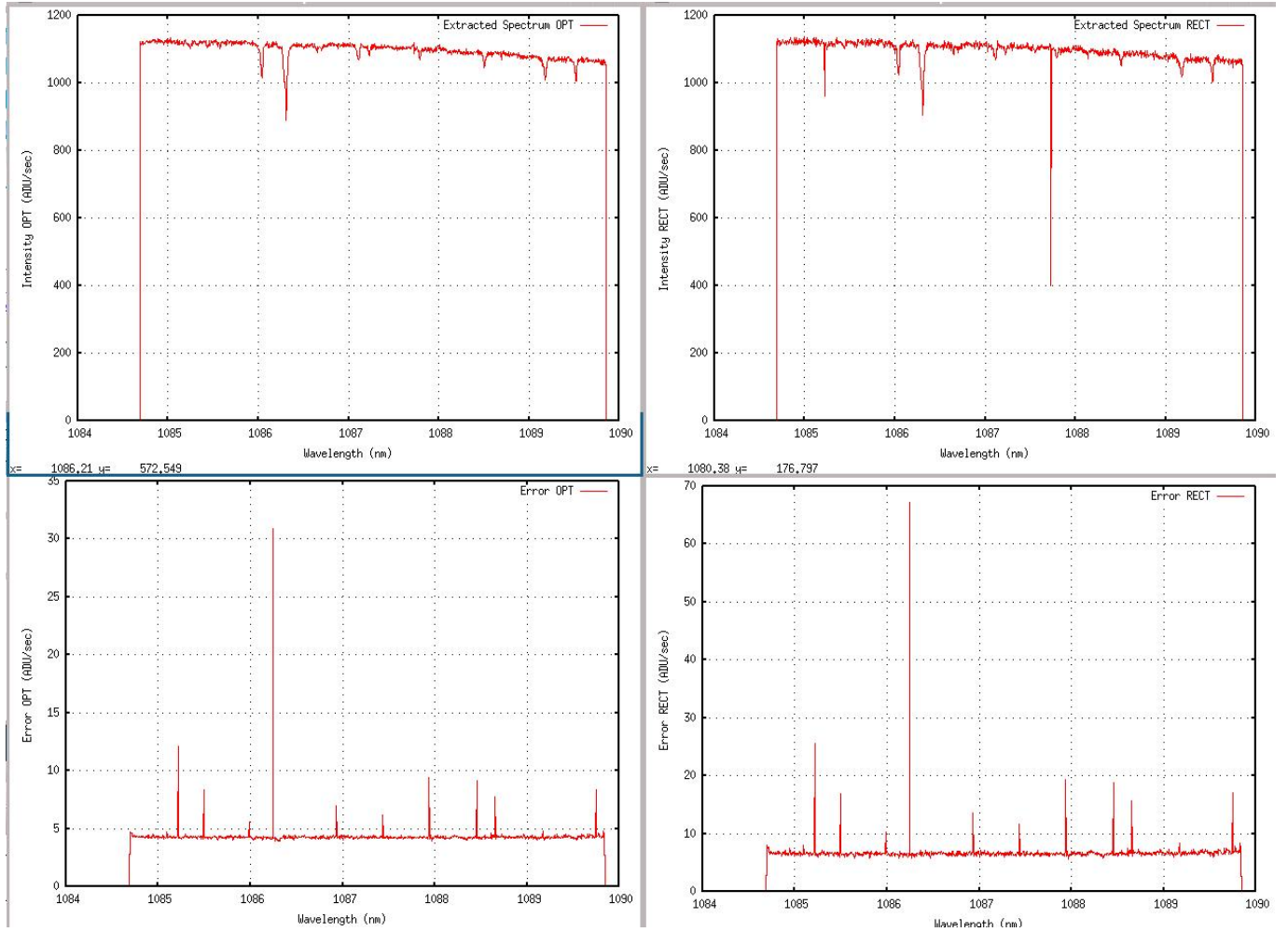


Figure 46: Dark subtraction, flatfielding, wavelength calibration, non-linearity correction, extraction and combination of simple nodding observations. In this case optimal extraction gives an improved S/N ratio.

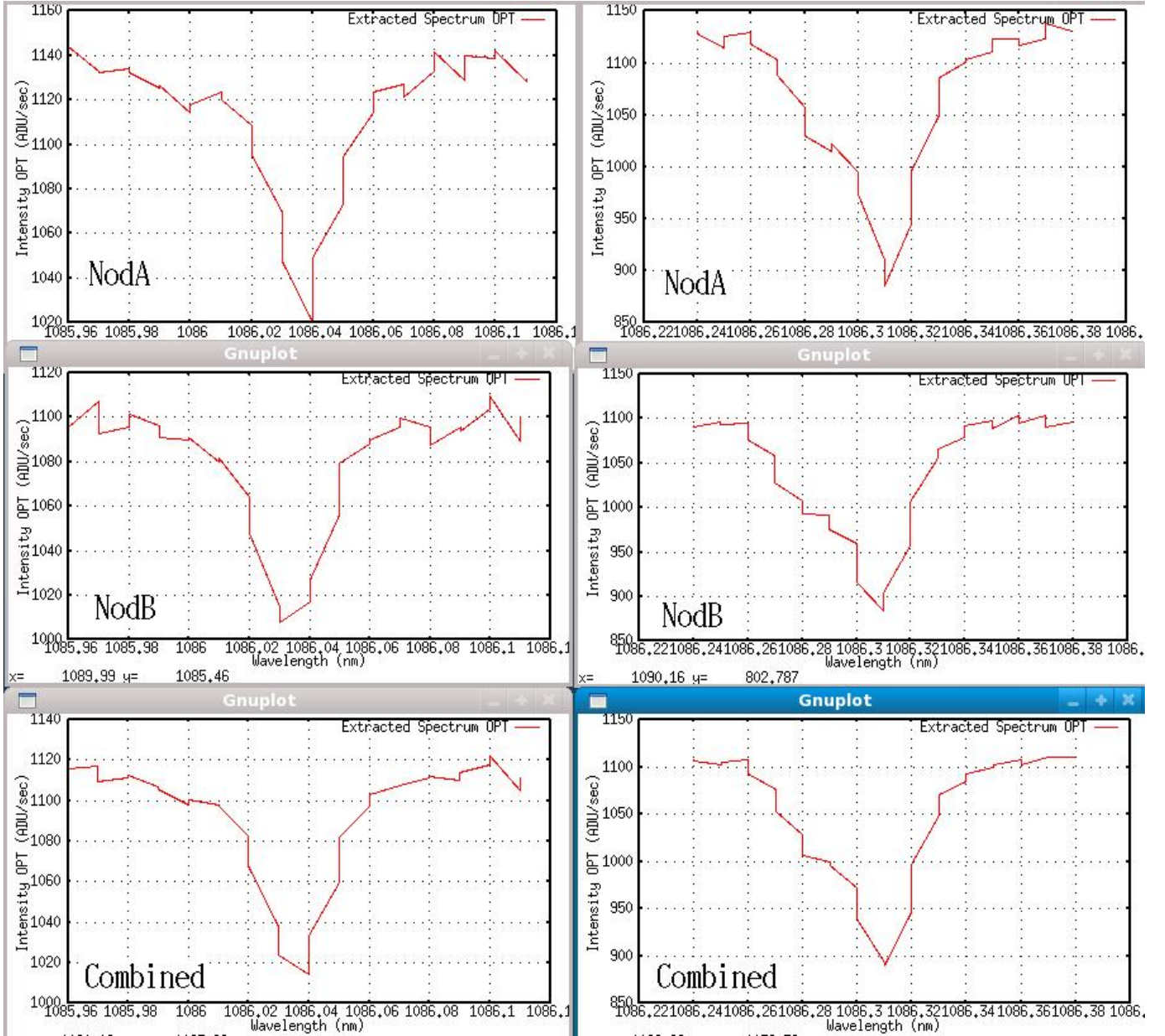


Figure 47: Extraction of NodA alone, NodB alone and the combined spectra. Slight shifts between NodA and NodB are apparent due to the fact that the wavelength calibration in v2.1.3 of the pipeline did not take into account slight differences in the solution on different 'Y' parts of the detectors. **This issue was fixed in v2.3.1 of the pipeline which can apply the derived solution at different Y-positions on the detectors - see Fig. 39 and Sect. 8.6.3.**

8.8 Reducing CRIRES data semi-automatically

Appendix C contains an example of a csh script that can be used to semi-automatically associate CRIRES calibrations and science files and reduce them using `crires_spec_jitter`.

8.9 Telluric line correction

At present, telluric line correction is not possible within the CRIRES pipeline. One should either use a telluric standard star taken close in time and airmass or an atmospheric model such as the one provided by ESO, available at www.eso.org/observing/etc/skycalc/skycalc.htm. In the foreseeable future as part of Austria's in-kind contribution to ESO, a tool to remove telluric features may be released; see [here](#)

Note that the exposure time calculator <http://www.eso.org/observing/etc/> also includes an atmospheric model and should be consulted before your observations to check the position of telluric lines compared with stellar or interstellar features of interest. See Fig. 48 for an example.

8.10 Importing reduced data into IDL or IRAF

8.10.1 IDL

To read in CRIRES reduced data to IDL then one uses the `mrdfits` command. Example scripts are shown in Appendix A.

8.10.2 IRAF

To read in CRIRES reduced data to IRAF then one uses `STRFITS` within `STSDAS` followed by `TXDUMP` to write out an ascii-format file. See Appendix B.

8.11 Plotting reduced spectra

To plot the reduced spectra within the CRIRES pipeline then follow the procedure explained in Sect. 7.4, using the recipe `crires_util_plot`.

9 Utilities recipes

The CRIRES pipeline has a number of utilities recipes that allow reduction of spectra. Note that when using the utility recipes great care has to be taken when tagging the files. In particular, use of the utility recipes with `TAGGED` inputs in a `sof` file is depreciated (although see below). Typically the input FITS files should be passed on the command line, for example: `esorex crires_util_wlcalib a.fits lines.fits`. The support of `COMMAND.LINE` tag is a workaround for `gasgano`. **The order of the expected input parameters is IMPORTANT** and always documented in the man-pages, for example:

```
% esorex -man-page crires_util_combine
```

```
NAME
```

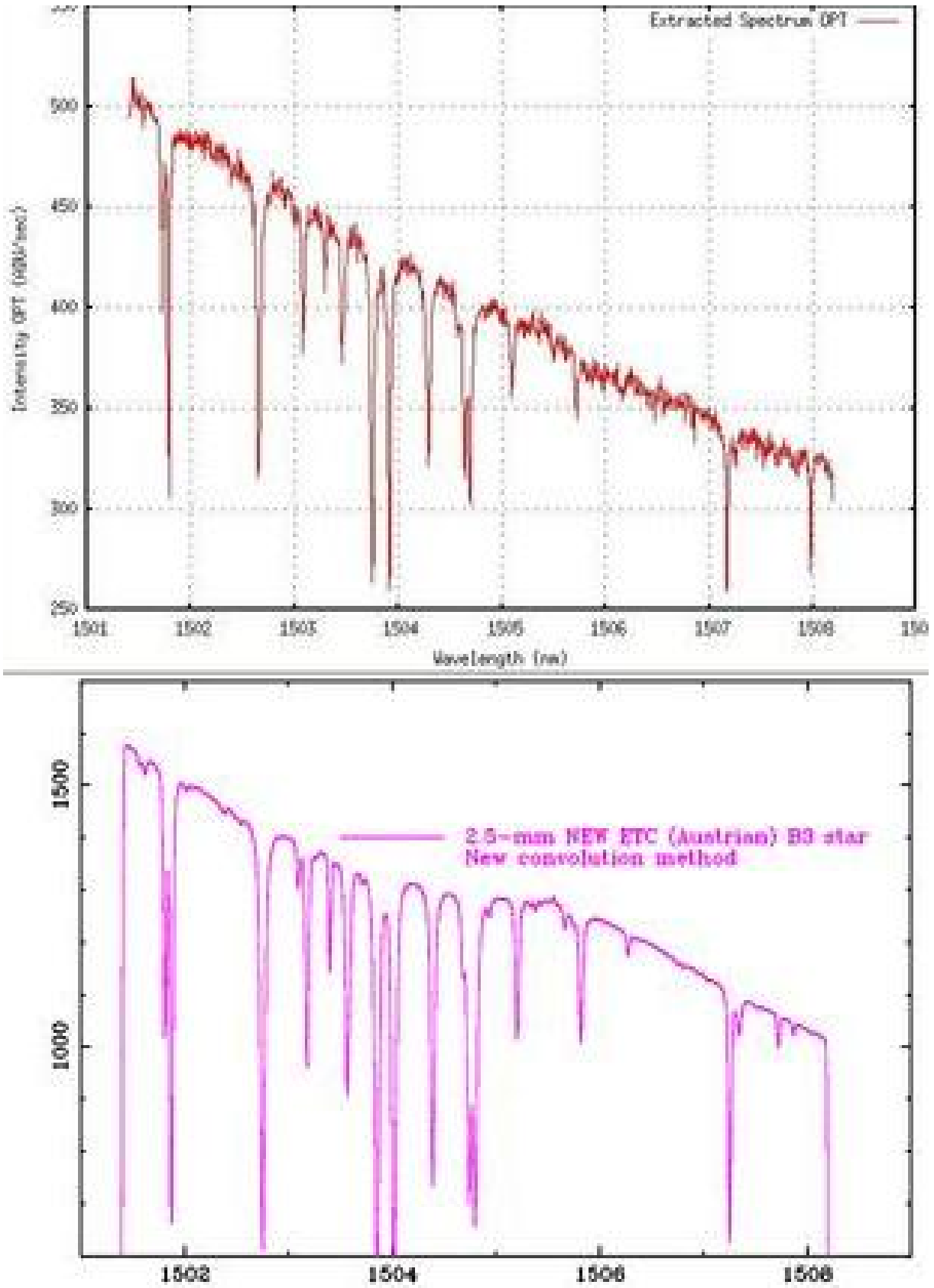


Figure 48: Output of the CRIRES exposure time calculator for a B3 star compared with actual observations of Eta Hy. Many telluric features caused by atmospheric absorption are visible in the spectra. The observations were taken with a water vapour column of around 2.5-mm which was the value used by the exposure time calculator (ETC).

```
crires_util_combine -- Images Combination tool
```

SYNOPSIS

```
esorex [esorex-options] crires_util_combine [crires_util_combine-options] sof
```

DESCRIPTION

```
crires_util_combine -- Images Combination tool
```

The files listed in the Set Of Frames (sof-file) must be tagged:

```
raw-file.fits OBS_DIR or
raw-file.fits OBS_DIR_JIT or
raw-file.fits CAL_DIR or
raw-file.fits CAL_DIR_JIT or
raw-file.fits OBS_OBJECT or
raw-file.fits OBS_SKY or
raw-file.fits OBS_GENERIC_OBJECT or
raw-file.fits OBS_GENERIC_SKY or
raw-file.fits OBS_NOD or
raw-file.fits OBS_NOD_JIT or
raw-file.fits CAL_NOD or
raw-file.fits CAL_NOD_JIT or
flat-file.fits CALPRO_FLAT or
bpm-file.fits CALPRO_BPM or
dark-file.fits CALPRO_DARK or
detlin-file.fits COEFFS_CUBE or
```

(snip)

```
% esorex -man-page crires_util_extract
```

```
***** ESO Recipe Execution Tool, version 3.9.0 *****
```

NAME

```
crires_util_extract -- Spectrum extraction routine
```

SYNOPSIS

```
esorex [esorex-options] crires_util_extract [crires_util_extract-options] sof
```

DESCRIPTION

This recipe accepts 1 parameter (optionally a 2nd one):

First parameter: the combined image.
(PRO TYPE = COMBINED)

Second parameter (optional): the contribution map.
(PRO TYPE = CONTRIB)

(snip)

9.1 Examples of the use of the utilities

Some examples of utility usage is described below.

9.1.1 `crires_util_combine`

This recipe to combine spectra has been tested with a SOF containing:

```
raw_image.fits      OBS_DIR
masterflat.fits     CALPRO_FLAT
badpixmap.fits      CALPRO_PBM
masterdark.fits     CALPRO_DARK
nonlinearity.fits   either DETLIN_A to _C (3 frames) or COEFFS_CUBE
                    (depending on the date of the raw_image)
```

9.1.2 `crires_util_extract`

This recipe can be used providing the files in the command line:

```
esorex --output-dir=outdir crires_util_extract combinedimage.fits contributionmap.fits
```

This recipe to extract spectra has been tested with a SOF containing:

```
combinedimage.fits  COMMAND_LINE
contributionmap.fits COMMAND_LINE
```

This recipe can also be used without a SOF, however for some purposes using a SOF is more convenient.

9.1.3 `crires_util_wlcalib`

This recipe can be used providing the files in the command line:

```
esorex --output-dir=outdir crires_util_wlcalib  extractedspectrum.fits
                                                referencelinecatalog.fits.fits
```

This wavelength-calibration recipe has been tested with a SOF containing:

```
extractedspectrum.fits  COMMAND_LINE
referencelinecatalog.fits COMMAND_LINE
```

9.1.4 `crires_util_wlassign`

This recipe can be used providing the files in the command line:

```
esorex --output-dir=outdir crires_util_wlassign extractedspectrum.fits wlcalib.fits
```

It can as well be used with a SOF providing the files (and using the `COMMAND_LINE` tag) but make sure that the order is as shown in the help file.

9.1.5 crires_util_genlines

This recipe is used to create a reference line catalogue. It is not clear (to be confirmed) if the recipe works through the command line. Using a SOF only one entry is needed:

```
asciilinetable.txt    CATALOG
```

10 Limitations and problems

The CRIRES pipeline (and the calibration data on which it works), suffers from a number of limitations, some of which are noted below.

10.1 Wavelength calibration issues

10.1.1 Limited accuracy in the ThAr and OH line catalogues

Until the release of CRIRES pipeline 2.3.1, the ThAr and OH catalogues used by the CRIRES pipeline (fits and ascii tables) only had an accuracy of two decimal places, for example 1265.27. This was fixed in version 2.3.1 of the pipeline.

10.1.2 Too few ThAr lines

In order to obtain a good wavelength solution with CRIRES, you should have at least two or three ThAr lines visible on each detector, preferably well spaced, at least for a 2nd order polynomial. Otherwise the results of the solution will likely not be trustworthy. Obviously this is not a problem of the pipeline but with the wavelength calibration source itself. Sometime during P93 it is hoped to include a UNe lamp in CRIRES that will solve a lot of the current issues at a stroke due to the high density of lines and large wavelength coverage. Typically UNe has 2–5 times as many lines as ThAr in the range 1 to 2.5 microns.

10.1.3 False identification of ThAr line features

The scratch on detector number 2 plus other features in the raw frames are often misinterpreted by the pipeline as being ThAr features. Masking of these may be performed before the wavelength calibration is performed, otherwise a degraded solution could result. One should check the output of the `crires_spec_wave` recipe to check that detected lines actually match up with known features.

10.2 Problems in tracing

If the Signal to Noise ratio is low, or if the spectrum just contains emission lines with no continuum, the extraction of the spectra will likely fail.

10.3 IRAF issues

Raw CRIRES slit viewer images and spectra can be displayed and analysed using `iraf` and `ds9`, for example the first four `iraf` commands will estimate the statistics of the four detectors

for a dark frame, with the fifth and sixth used to display and show the fits header of a CRIRES slit viewer image:

```
cl> imstat CRIRES_SPEC_DARK108_0077.fits[1]
#          IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
  CRIRES_SPEC_DARK108_0077.fits[1]      524288      1.711      34.88      -575.2      7111.
cl> imstat CRIRES_SPEC_DARK108_0077.fits[2]
#          IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
  CRIRES_SPEC_DARK108_0077.fits[2]      524288      0.6321      62.17      -416.4      8514.
cl> imstat CRIRES_SPEC_DARK108_0077.fits[3]
#          IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
  CRIRES_SPEC_DARK108_0077.fits[3]      524288      0.5176      29.52      -376.3      6682.
cl> imstat CRIRES_SPEC_DARK108_0077.fits[4]
#          IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
  CRIRES_SPEC_DARK108_0077.fits[4]      524288      3.9      182.      -1948.      16182.
```

```
cl> display Cries_sv_ngc6253_068_0001.fits 1 fi+
z1=45. z2=1833.705
cl> imhead Cries_sv_ngc6253_068_0001.fits lo+
```

```
Cries_sv_ngc6253_068_0001.fits[1024,1024][real]: TEST
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [1024,1024], length of user area 21668 s.u.
Created Thu 06:40:33 08-Mar-2012, Last modified Thu 02:28:55 19-Apr-2012
Pixel file "Cries_sv_ngc6253_068_0001.fits" [ok]
EXTEND   =                               T / FITS Extension may be present
ORIGIN   = 'ESO-Paranal'                  / Source of the file
DATE     = '2012-03-08T09:40:32.7467' / Date the file was written
TELESCOP = 'ESO-VLT-U1'                   / Telescope name
INSTRUME = 'CRIRES'                       / Instrument name
OBJECT   = 'TEST'                         / Content description
RA       =          254.775705             / 16:59:06.1 RA (J2000) pointing (deg)
DEC      =          -52.71581              / -52:42:56.9 DEC (J2000) pointing (deg)
EQUINOX  =          2000.                  / Standard FK5 (years)
RADECSYS = 'FK5'                          / Coordinate reference frame
EXPTIME  =          10.0000000 / Integration time
MJD-OBS  =          55994.40302817 / Obs start 2012-03-08T09:40:21.634
<snip>
```

Additionally, some reduced calibration frames can be displayed and manipulated:

```
cl> display OBS_WL_MAP_IMA_WithoutCals_100s_A.fits[1]
frame to be written into (1:16) (1):
z1=4034.347 z2=4054.035

cl> imstat SpecJit_1087p4_PipeDefaults_STD_BGD_MAP_IMA.fits[1]
#          IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
  SpecJit_1087p4_PipeDefaults_STD_BGD_MAP_IMA.fits[1]      524288      0.6774      0.4675
```

```
cl> imarith SpecJit_1087p4_PipeDefaults_STD_BGD_MAP_IMA.fits[1] * 100.0
      SpecJit_1087p4_PipeDefaults_STD_BGD_MAP_IMA_Times_100_Detector1.fits
```

Problems can occur when plotting the final reduced spectra;

```
on> splot SpecJit_1087p4_PipeDefaults_STD_EXTRACT_SENS_TAB.fits
ERROR: MWCS: dimension mismatch (mw_open)
```

One solution to this is to read in the data into a table format as explained in Appendix 1.

The following iraf-related issues that the ESO pipelines have was taken from:

<http://www.eso.org/observing/dfo/quality/CRIRES/ServiceMode/ServiceMode.html> which should be considered the main reference.

- **Filename Length.** To display or manipulate the FITS files with older versions of IRAF (before 2.11), you can: - copy these FITS files to your hard-disk and rename them with filenames ≤ 32 characters in length; - create symbolic links with filenames ≤ 32 characters in length to your DVD files.
- **Header Interpretation.** ESO FITS files use the ESO HIERARCH FITS keyword extensions standard to all ESO telescopes. Note that IRAF treats all ESO HIERARCH header lines as COMMENT lines, i.e. IRAF and IDL cannot automatically interpret the information provided in ESO HIERARCH header lines. The problem may be solved using the tool `hierarch28`. Find information about this tool at <http://archive.eso.org/saft/>.

```
hierarch28 -g
jsmoker> hierarch28 SpecJit_1087p4_PipeDefaults_STD_EXTRACT_SENS_TAB.fits
```

```
*** hierarch28
```

searching SpecJit_1087p4_PipeDefaults_STD_EXTRACT_SENS_TAB.fits and replacing the following keywords:

```
[UTC] =>[UT]
[LST] =>[ST]
[RA] =>[RA]
[DEC] =>[DEC]
[HIERARCH ESO TEL AIRM START] =>[AIRMAS]
[HIERARCH ESO DPR TYPE] =>[IMAGETYP]
[HIERARCH ESO INS FILT1 NAME] =>[FILTER1]
[HIERARCH ESO INS SLIT2 NAME] =>[SLIT]
[HIERARCH ESO INS GRIS1 NAME] =>[GRISM]
[HIERARCH ESO INS GRAT NAME] =>[GRAT]
[HIERARCH ESO INS GRAT1 NAME] =>[GRAT1]
[HIERARCH ESO INS GRAT2 NAME] =>[GRAT2]
[HIERARCH ESO INS GRAT WLEN] =>[WLEN]
[HIERARCH ESO INS GRAT1 WLEN] =>[WLEN1]
[HIERARCH ESO INS GRAT2 WLEN] =>[WLEN2]
[HIERARCH ESO INS GRAT ORDER] =>[ORDER]
```

One can also use the following idl code provided by Alain Smette:

```
# asmette 2010-03-21 v1.0
function get_hierarch,h,keyword
    keyname = 'HIERARCH ESO',
    strput,keyname,keyword,13

    keyname = strtrim(keyname)
    res      = strpos(h,keyname)
    pos      = where(res eq 0)
    if pos[0] eq -1 then begin
        return,-1
    endif
    line     = h[pos]
    pos      = strpos(line,"'")

    if pos[0] gt 0 then begin
        pos    = strpos(line,"'",32)
        value  = strmid(line, 32,pos[0]-32)
    endif else begin
        pos    = strpos(line,"/")
        poseq  = strpos(line,"=")+1
        value  = float(strmid(line[0], poseq[0],pos[0]-poseq[0]))
    endelse
    return,value[0]
end
```

- **RA, DEC.** Please note that the RA and DEC keywords are recorded in degrees. To translate these keywords so that they can be used by IRAF you have to use the `asthedit` task in the `noao.astutil` package. The help file for this task gives an example of how to translate the ESO format to the IRAF format.

10.4 Correcting for non-linearity

The non-linearity correction for CRIRES has its limitations, in great part due to the detector itself. Figure 49 shows an example of the use of the CRIRES flatfield with the gas-cell inserted for DIT=100-s, NDI=1 and DIT=2-s, NDI=50. These data can be used to check the efficacy of the non-linearity correction by looking at the depth of the absorption lines in the spectra. More information can be obtained by contacting crires@eso.org. Recall that although it has limitations, one should always apply the non-linearity fits files as described in Sect. 3.3.5.

10.5 Problems in running gasgano

Following are a handful of problems that users may encounter when running `gasgano` and how to avoid them.

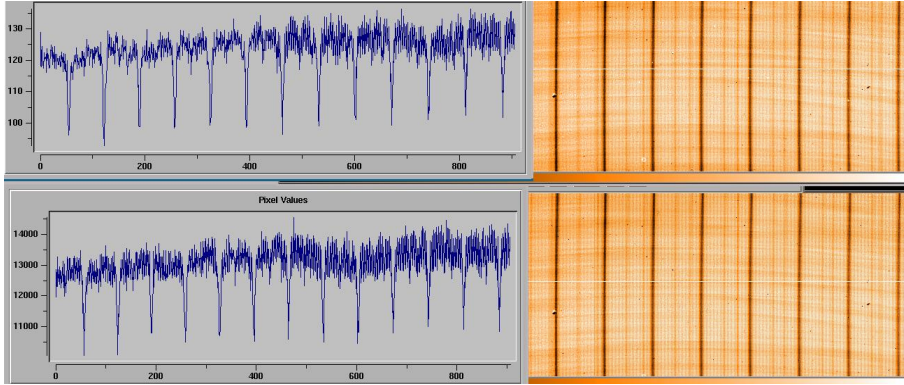


Figure 49: Raw CRIRES images with DIT=2-s, NDIT=50 (top plot) and DIT=100-s, NDIT=1 (bottom plot).

Filename	DIT (s)	NDIT	Count Det 3 (ADU)
CRIRE.2012-04-17T03:31:14.065.fits	1	100	80
CRIRE.2012-04-17T04:23:18.166.fits	2	50	190
CRIRE.2012-04-17T04:20:52.616.fits	3	34	300
CRIRE.2012-04-17T04:11:57.389.fits	4	25	400
CRIRE.2012-04-17T04:14:18.624.fits	5	20	500
CRIRE.2012-04-17T04:16:33.561.fits	10	10	1000
CRIRE.2012-04-17T04:18:44.152.fits	20	5	1900
CRIRE.2012-04-17T04:20:52.616.fits	40	3	3800
CRIRE.2012-04-17T04:23:18.166.fits	50	2	4800
CRIRE.2012-04-17T04:25:20.496.fits	67	1	6200
CRIRE.2012-04-17T04:26:47.827.fits	100	1	9000
CRIRE.2012-04-17T04:28:48.799.fits	120	1	11000

Table 3: DIT and NDIT combinations for test of the CRIRES non-linearity correction.

- Gasgano will not start with CPL initialisation failed:

```
gasgano      gasgano_227 gasgano_230 gasgano_253 gasgano_260
[jsmoker@localhost ~]$ gasgano &
[1] 19723
[jsmoker@localhost ~]$ Lost output message: 10:52:04 [WARNING] CPL-messaging
                                initialization failed:
Lost error message: 10:52:04 [ ERROR ] Dumping all 1 error(s):
Lost error message: 10:52:04 [ ERROR ]      [1/1] 'File cannot be created'
                                (8) at cpl_msg_set_log_level:cpl_msg.c:874
```

Solution: Likely you do not have write access for the current directory. Change directory to somewhere where you can create files.

- Gasgano hangs when adding new files:

Solution: You may have .fits.gz files in the tree. Either move them somewhere else or gunzip them.

- Recipe fails because frames not classified:

Occasionally you may see an "UNDEF" in the "Classification" column of gasgano recipe window (e.g. Fig. 44). This either means that there is a problem with the FITS headers (DPR.CATG, DPR.TECH and/or DPR.TYPE), or that the "rules" file used to classify the frames is not loaded and/or is faulty. You can still force gasgano to do a reduction by entering the classification by hand, for example click on "UNDEF" and change to "CALPRO_FLAT" if you are sure that image is a master flatfield.

- Telescope keywords not present in reduced files.

Solution: Make sure that the first file in your .sof file is a science file. If it is a calibration then the TEL keywords will not be present in the output file because they are not present in the calibration.

10.6 Cannot find recipe files

Sometimes you may have several versions of the CRIRES pipeline on your computer. To select a particular set of recipes just add the following to your esorex command:

```
esorex --recipe-dir=/path/crre-kit-2.3.0/crre-2.3.0/recipes/.libs
```

To check which recipes gasgano is looking at go to File → Preferences → Recipe Configuration from the gasgano main panel.

10.7 Problems in running esorex

A list of frequently-asked questions is available here: <http://www.eso.org/sci/software/cpl/faq.html>

A Example IDL script to read in reduced data and produce ascii-format output files

If we have IDL and an input file called `Reduced_wl_map_filelist.ascii` which contains a list of the spectra with product category CRIRES OBS_EXTRACT_WL_TAB reduced files produced by the pipeline, combined with output names for each of the 4 detectors.

```
% cat Reduced_wl_map_filelist.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_001s_A.fits   OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch1.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch2.ascii OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch3.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch4.ascii   (all on one line)
OBS_EXTRACT_WL_TAB_WithAllCals_002s_A.fits   OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch1.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch2.ascii OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch3.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch4.ascii   (all on one line)
OBS_EXTRACT_WL_TAB_WithAllCals_004s_A.fits   OBS_EXTRACT_WL_TAB_WithAllCals_004s_A_ch1.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_004s_A_ch2.ascii OBS_EXTRACT_WL_TAB_WithAllCals_004s_A_ch3.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_004s_A_ch4.ascii   (all on one line)
OBS_EXTRACT_WL_TAB_WithAllCals_005s_A.fits   OBS_EXTRACT_WL_TAB_WithAllCals_005s_A_ch1.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_005s_A_ch2.ascii OBS_EXTRACT_WL_TAB_WithAllCals_005s_A_ch3.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_005s_A_ch4.ascii   (all on one line)
OBS_EXTRACT_WL_TAB_WithAllCals_010s_A.fits   OBS_EXTRACT_WL_TAB_WithAllCals_010s_A_ch1.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_010s_A_ch2.ascii OBS_EXTRACT_WL_TAB_WithAllCals_010s_A_ch3.ascii
OBS_EXTRACT_WL_TAB_WithAllCals_010s_A_ch4.ascii   (all on one line)
```

Then we can read in the fits files and write out the individual output files by doing the following:

```
[jsmoker@pc015042 ReducedData]$ idl
IDL Version 8.1 (linux x86_64 m64). (c) 2011, ITT Visual Information Solutions

IDL> .r CRIRES_Convert_Fits_To_Ascii.pro
% Compiled module: CRIRES_CONVERT_FITS_TO_ASCII.

IDL> CRIRES_Convert_Fits_To_Ascii
```

This will produce output files called `*ch1.ascii`, `*ch2.ascii`, `*ch3.ascii` and `*ch4.ascii`. The `CRIRES_Convert_Fits_To_Ascii` is listed below. The code is very simple but works.

```
pro CRIRES_Convert_Fits_To_Ascii
;
; Basic code that reads in reduced OBS_EXTRACT_WL_TAB fits files produced by the CRIRES pipeline
; and produces ascii-format files with columns for the wavelength and extracted optimal
; extraction.
;
; The directory dir='/home/jsmoker/CRIRES/Projects/Linearity/ReducedData/' should be changed
; in most cases.
; The input file Reduced_wl_map_filelist.ascii contains lines that look like the following;
; OBS_EXTRACT_WL_TAB_WithAllCals_001s_A.fits OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch1.ascii
; OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch2.ascii OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch3.ascii
; OBS_EXTRACT_WL_TAB_WithAllCals_001s_A_ch4.ascii   (all one line)
```



```

; OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch1.ascii OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch2.ascii
; OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch3.ascii OBS_EXTRACT_WL_TAB_WithAllCals_002s_A_ch4.ascii
; (all on one line)
; To see the structure of the file with column names, do the following
; help, ch1, /structure
; read in input and output files from a list
;
; the following directory should be changed
;
dir='/home/jsmoker/CRIRES/Projects/Linearity/ReducedData/'
fname = 'Reduced_wl_map_filelist.ascii'

readcol, dir+fname, datafileIN, datafileOUT_1, datafileOUT_2, datafileOUT_3, datafileOUT_4,
      format='A,A,A,A,A,A', comment='{', /silent
i = 0
repeat begin
  j=2
  print, "Reading in CH1 data:",datafileIN[i]
  openw,lun,datafileOUT_1[i],/get_lun
  datafileTEMP = datafileIN[i]
  header = headfits(datafileTEMP)
  ch1 = mrdfits(datafileTEMP,1,head)
  ch2 = mrdfits(datafileTEMP,2,head)
  ch3 = mrdfits(datafileTEMP,3,head)
  ch4 = mrdfits(datafileTEMP,4,head)

  repeat begin
    printf,lun,ch1[j].wavelength,ch1[j].extracted_opt, format='(f22.5,f22.5)'
    j = j+1
  endrep until j eq 1022
  free_lun, lun
  i=i+1
  print,'Counter i equals ',i
endrep until i eq 34

i = 0
repeat begin
  j=2
  print, "Reading in CH2 data: ",datafileIN[i]
  openw,lun,datafileOUT_2[i],/get_lun
  datafileTEMP = datafileIN[i]
  header = headfits(datafileTEMP)
  ch1 = mrdfits(datafileTEMP,1,head)
  ch2 = mrdfits(datafileTEMP,2,head)
  ch3 = mrdfits(datafileTEMP,3,head)
  ch4 = mrdfits(datafileTEMP,4,head)

  repeat begin
    printf,lun,ch2[j].wavelength,ch2[j].extracted_opt, format='(f22.5,f22.5)'
    j = j+1
  endrep until j eq 1022

```

```
    free_lun, lun
    i=i+1
endrep until i eq 34

i = 0
repeat begin
    j=2
    print, "Reading in CH3 data: ",datafileIN[i]
    openw,lun,datafileOUT_3[i],/get_lun
    datafileTEMP = datafileIN[i]
    header = headfits(datafileTEMP)
    ch1 = mrdfits(datafileTEMP,1,head)
    ch2 = mrdfits(datafileTEMP,2,head)
    ch3 = mrdfits(datafileTEMP,3,head)
    ch4 = mrdfits(datafileTEMP,4,head)

    repeat begin
        printf,lun,ch3[j].wavelength,ch3[j].extracted_opt, format='(f22.5,f22.5)'
        j = j+1
    endrep until j eq 1022
    free_lun, lun
    i=i+1
endrep until i eq 34

i = 0
repeat begin
    j=2
    print, "Reading in CH4 data:",datafileIN[i]
    openw,lun,datafileOUT_4[i],/get_lun
    datafileTEMP = datafileIN[i]
    header = headfits(datafileTEMP)
    ch1 = mrdfits(datafileTEMP,1,head)
    ch2 = mrdfits(datafileTEMP,2,head)
    ch3 = mrdfits(datafileTEMP,3,head)
    ch4 = mrdfits(datafileTEMP,4,head)

    repeat begin
        printf,lun,ch4[j].wavelength,ch4[j].extracted_opt, format='(f22.5,f22.5)'
        j = j+1
    endrep until j eq 1022
    free_lun, lun
    i=i+1
endrep until i eq 34

end
```

B Example IRAF script to read in reduced data and produce ascii-format output files

The original script was written by Ronald Mennickent Cid and then modified into a procedure by Pedro Figuera. We gratefully acknowledge their help. The current version only provides output in pixel space.

If we have IRAF and wish to read in a CRIRES reduced spectrum with category CRIRES OBS_EXTRACT_WL_TAB produced by the pipeline, we need to use `strfits` and `tdump` in the following way;

```
#####
# start procedure with
# task convESO ="/home/pfigueira/iraf_data/scripts/DRACS/src/convESO.cl"
#####

string name_spectrum          {prompt="Name of spectrum to be converted"}

begin
strfits.fits_file=name_spectrum/"_1D01.fits"
strfits.file_list=""
strfits.iraf_file=name_spectrum/"_1D"
strfits.template="none"
strfits.long_he=no
strfits.short_h=yes
strfits.datatyp="default"
strfits.blank=0.
strfits.scale=yes
strfits.xdimtog=yes
strfits.olderaf=no
strfits.force=yes
strfits.offset=0

strfits

#dump into ascii's
tdump.table=name_spectrum/"_1D01.tab"
tdump > "Extracted_1D01.ascii"

tdump.table=name_spectrum/"_1D02.tab"
tdump > "Extracted_1D02.ascii"

tdump.table=name_spectrum/"_1D03.tab"
tdump > "Extracted_1D03.ascii"

tdump.table=name_spectrum/"_1D04.tab"
tdump > "Extracted_1D04.ascii"

# retrieve only wavelength and extracted spectra flux
!cat "Extracted_1D01.ascii" | grep -iv e | awk '{printf "%20.5f%20.5f\n", $8, $2}'
    > "Extracted_1D01_Wlen_ExtOPT_Err.ascii"          (all one line)
!cat "Extracted_1D02.ascii" | grep -iv e | awk '{printf "%20.5f%20.5f\n", $8, $2}'
```

```

    > "Extracted_1D02_Wlen_ExtOPT_Err.ascii"          (all one line)
!cat "Extracted_1D03.ascii" | grep -iv e | awk '{printf "%20.5f%20.5f\n", $8, $2}'
    > "Extracted_1D03_Wlen_ExtOPT_Err.ascii"          (all one line)
!cat "Extracted_1D04.ascii" | grep -iv e | awk '{printf "%20.5f%20.5f\n", $8, $2}'
    > "Extracted_1D04_Wlen_ExtOPT_Err.ascii"          (all one line)

#writing the results in the form of image files
rspectext.dtype="linear"
rspectext.crval1=1.
rspectext.cdelt1=1.
rspectext.input="Extracted_1D01_Wlen_ExtOPT_Err.ascii"
rspectext.output=name_spectrum/"_1D_det01"
rspectext

rspectext.input="Extracted_1D02_Wlen_ExtOPT_Err.ascii"
rspectext.output=name_spectrum/"_1D_det02"
rspectext

rspectext.input="Extracted_1D03_Wlen_ExtOPT_Err.ascii"
rspectext.output=name_spectrum/"_1D_det03"
rspectext

rspectext.input="Extracted_1D04_Wlen_ExtOPT_Err.ascii"
rspectext.output=name_spectrum/"_1D_det04"
rspectext

#copying the headers so that information is not lost
mkheader(images=name_spectrum/"_1D_det01", headers=name_spectrum)
mkheader(images=name_spectrum/"_1D_det02", headers=name_spectrum)
mkheader(images=name_spectrum/"_1D_det03", headers=name_spectrum)
mkheader(images=name_spectrum/"_1D_det04", headers=name_spectrum)

#removing temporary files
!rm crires_spec_jitter_extracted_000?_1D0* Extracted_1D0*

end

```

The above creates a file called *.hhh and *.tab (note the force=yes to force conversion from fits files and oldiraf=no). Now we need to write this out in ascii format using the `ttools` package.

```

cl> tdump OBS_EXTRACT_WL_TAB_WithAllCals_120s_A01.tab >
      OBS_EXTRACT_WL_TAB_WithAllCals_120s_A01.ascii (all on one line)

```

This produces an ascii file with nine columns, including the wavelength and extracted counts for optical and rectangular, amongst others.

```

% cat OBS_EXTRACT_WL_TAB_WithAllCals_120s_A01.ascii
Extracted_RECT    D          %25.16g    ""

```

```

Extracted_OPT      D          %25.16g  ""
Error_RECT         D          %25.16g  ""
Error_OPT          D          %25.16g  ""
Weight_Norm_RECT   D          %25.16g  ""
Weight_Norm_OPT    D          %25.16g  ""
Background_noise   D          %25.16g  ""
Wavelength         D          %25.16g  ""
Wavelength_model   D          %25.16g  ""
HIERARCH t 'ESO'   QC FWHM DIFF = -23.4195905544356
HIERARCH t 'ESO'   QC FWHM DIFF = -23.4195905544356
    4738.449440002442      4738.449616523445      2.381331792048803      2.381331880760329
        41.      34259.43634033203      0.      4034.38590909091
    (all on one line)
    4732.57120513916      4732.571381441183      2.379854258206329      2.379854346862813
        41.      34259.43634033203      0.      4034.405363636364
    (all on one line)
    4744.603256225586      4744.603432975838      2.382877806448656      2.382877895217775
        41.      34259.43634033203      0.      4034.424818181818
    (all on one line)
(snip)

```

If one wanted to select just the wavelength and optimally extracted columns one could then do;

```

% cat OBS_EXTRACT_WL_TAB_WithAllCals_120s_A01.ascii | grep -iv e | '{printf "%20.5f%20.5f\n", $8
> OBS_EXTRACT_WL_TAB_WithAllCals_120s_A01_Wlen_ExtOPT_Err.ascii (all one line)

```

Thanks to Ronald Mennickent Cid for the above information.

C Script to associate science and calibration data, create master frames and extract using `crires_spec_jitter`

Given appropriate calibration and science data (downloaded from the ESO archive using CalSelector), a script has been written that associates CRIRES calibrations with science files, uses `esorex` to produce master calibrations and then uses these master calibrations plus the science data to extract CRIRES nodding data. Please write to crires@eso.org if you either would like to use the script or have a better script and are willing to share it.

___oOo___