



# EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

## VERY LARGE TELESCOPE

### Detector monitoring recipes User Manual

VLT-MAN-ESO-19500-4846

Issue 1.3.12

Date 2023-03-14

Prepared: .....  
Name ..... Date ..... Signature .....

Approved: .....  
Name ..... Date ..... Signature .....

Released: .....  
Name ..... Date ..... Signature .....

This page was intentionally left blank

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 3 of 37                |

### Change record

| Issue/Rev. | Date       | Section/Parag. affected | Reason/Initiation/Documents/Remarks                                  |
|------------|------------|-------------------------|--|
| 0.1        | 19/03/2007 | All                     | First draft  |
| 0.2        | 19/03/2007 | 10,11                   | Updated linearity-gain   |
| 0.3        | 01/06/2007 | 10,11                   | Added RON-BIAS-DARK  |
| 0.4        | 21/04/2008 | all                     | Updated recipe-io and algorithms                                     |
| 0.5        | 08/09/2008 | all                     | Updated recipe-io and algorithms                                     |
| 0.6        | 22/09/2009 | all                     | Revised (removed all unnecessary parts)                              |
| 0.7        | 11/11/2009 | all                     | Clarified FPN,gain and linearity tables                              |
| 0.8        | 20/05/2010 | all                     | Clarified saturation_limit parameter                                 |
| 0.9        | 14/06/2010 | all                     | Removed not supported recipe/algorithms                              |
| 0.9.5      | 09/07/2010 | sec 4                   | To address recipe usage for operations                               |
| 0.9.6      | 09/07/2010 | sec 4                   | Included comments from QC-Garching                                   |
| 1.0.0      | 15/10/2010 |                         | First release as a detmon pipeline                                   |
| 1.0.2      | 29/11/2010 |                         | Few updates  |
| 1.0.3      | 17/12/2010 |                         | new parameter <i>coeffs_cube_split</i> ;<br>"Empty Frames" paragraph |
| 1.0.8      | 02/03/2011 |                         | ocam suport  |
| 1.2.2      | 31/05/2012 |                         | Update to CPL 6.1.1  |
| 1.2.3      | 21/04/2013 |                         | Update to CPL 6.3  |
| 1.2.7      | 23/03/2014 |                         | Update to CPL 6.4  |
| 1.2.7      | 11/08/2014 |                         | Update to CPL 6.5.1  |
| 1.3.0      | 24/02/2015 |                         | Update to CPL 6.6  |

This page was intentionally left blank

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 5 of 37                |

## Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Pipeline Recipes Interfaces</b>   | <b>7</b>  |
| 1.1      | detmon_ir_lg . . . . .   | 7         |
| 1.1.1    | Input . . . . .  | 7         |
| 1.1.2    | Output . . . . .   | 7         |
| 1.1.3    | Quality control . . . . .  | 9         |
| 1.1.4    | Parameters . . . . .   | 10        |
| 1.2      | detmon_opt_lg . . . . .  | 11        |
| 1.2.1    | Input . . . . .  | 11        |
| 1.2.2    | Output . . . . .   | 11        |
| 1.2.3    | Quality control . . . . .  | 13        |
| 1.2.4    | Parameters . . . . .   | 14        |
| 1.3      | Multi-region recipes . . . . .   | 16        |
| 1.3.1    | Input . . . . .  | 16        |
| 1.3.2    | Output . . . . .   | 16        |
| <b>2</b> | <b>Algorithms</b>  | <b>18</b> |
| 2.1      | Basic Algorithms . . . . .   | 18        |
| 2.1.1    | Histogram computation . . . . .  | 18        |
| 2.1.2    | Iteratives clean mean and sigma computation . . . . .                              | 18        |
| 2.1.3    | Detector contamination computation: detmon_nir_contamination, NIR domain . . . . . | 18        |
| 2.2      | General Algorithms . . . . .   | 18        |
| 2.2.1    | PTC and MED Algorithms for Gain computation . . . . .                              | 19        |
| 2.2.2    | Detector linearity computation (IR): irplib_detmon_nir_linear . . . . .            | 20        |
| 2.2.3    | Detector linearity computation (optical): irplib_detmon_opt_linear . . . . .       | 22        |
| 2.2.4    | Fixed Pattern Noise determination . . . . .  | 22        |
| 2.3      | Recipe Algorithms . . . . .  | 23        |
| 2.3.1    | detmon_ir_lg, detmon_opt_lg . . . . .  | 23        |
| <b>3</b> | <b>Usage of common detlin recipe across instruments</b>                            | <b>26</b> |
| 3.1      | How to execute an instrument specific and detmon pipelines . . . . .               | 26        |

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 6 of 37                |

|       |  |    |
|-------|--|----|
| 3.2   | Handling parameter setting priorities . . . . .                                  | 26 |
| 3.2.1 | How to activate instrument specific and detmon pipelines . . . . .               | 27 |
| 3.2.2 | How to de-activate detmon pipelines . . . . .                                    | 27 |
| 3.2.3 | How to specify instrument-specific input parameters . . . . .                    | 28 |
| 3.2.4 | Processing linearity data with multi extensions . . . . .                        | 32 |
| 3.2.5 | Instrument specific configuration files . . . . .                                | 32 |
| 3.2.6 | How to keep instrument pipeline and common recipe aligned . . . . .              | 36 |
| 3.2.7 | How to write for COEFFS_CUBE product each extension in a separate file . . . . . | 37 |

|     |   |        |                        |
|-----|---|--------|------------------------|
| ESO | Detector monitoring recipes User Manual | Doc:   | VLT-MAN-ESO-19500-4846 |
|     |   | Issue: | Issue 1.3.12           |
|     |   | Date:  | Date 2023-03-14        |
|     |   | Page:  | 7 of 37                |

## 1 Pipeline Recipes Interfaces

In this section we provide for each recipe examples of the required input data. and corresponding frame TAGs as expected by the recipes provided by the detmon package.

An instrument pipeline using the corresponding detmon algorithms may rely on different tag values. In that case we remind the user to the corresponding pipeline user manual and on line man page recipe documentation.

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

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

### 1.1 detmon\_ir\_lg

This recipe determines Linearity and Gain of detectors in the infrared domain.

The Linearity reduction part computes the detector response as a function of the pixel intensity, fitting this function to a polynomial whose coefficients help to determine when it becomes non linear.

The Gain reduction part computes the Gain of the detector.

Both reduction steps are merged in this recipe as both accept the same input interface. There are some variants to both reduction parts, controlled by the input parameters. See (reference to Algorithms) for details.

#### 1.1.1 Input

This recipe expects input frames classified either as ON\_RAW or as OFF\_RAW. Each instrument pipeline adopting detmon algorithms may change this value. Please refer to each instrument corresponding recipe man page.

If no modification is done to the default behaviour, there must be two frames tagged ON\_RAW and two frames tagged OFF\_RAW for each distinct DIT value found among input frames.

If the parameter **collapse** is set to TRUE, then the requirement for pairs of frames tagged OFF\_RAW disappears, and these frames are used in a different way.

The number of distinct DIT values among input frames must be at least **order** + 1 as well.

#### 1.1.2 Output

The following table lists all possible products of this recipe. Please note that not all of them are always created. Creation of some of these products depends on the value(s) of some input parameter(s).

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 8 of 37                |

| Default recipe filename           | PRO.CATG     | Short description      |
|-----------------------------------|--------------|------------------------|
| detmon_ir_lg_linearity_table.fits | DET_LIN_INFO | Linearity Table        |
| detmon_ir_lg_gain_table.fits      | GAIN_INFO    | Gain Table             |
| detmon_ir_lg_coeffs_cube.fits     | COEFFS_CUBE  | Coefficients cube      |
| detmon_ir_lg_bpm.fits             | BP_MAP_NL    | Bad pixel map          |
| detmon_ir_lg_diff_flat_i.fits     | DIFF_FLAT    | Difference images      |
| detmon_ir_lg_autocorr_i.fits      | AUTOCORR     | Autocorrelation images |

The last two products are generated only if **autocor** is set to TRUE.

If no modification is done to the default behaviour, this recipe produces only the first four products in the list.

Further details about conditions for product creation can be found along this section.

Both Linearity and Gain Tables are created in all possible execution cases.

### Content of the Linearity Table

For each distinct DIT value, this table logs:

| Name     | Description   |
|----------|---|
| DIT      | Detector Integration Time in s                            |
| MED      | Median intensity of $[(on_1 - off_1) + (on_2 - off_2)]/2$ |
| MEAN     | Mean intensity of $[(on_1 - off_1) + (on_2 - off_2)]/2$   |
| MED_DIT  | Ratio MED / DIT   |
| MEAN_DIT | Ratio MEAN / DIT  |
| ADL      | Column DIT multiplied by mean value of column MED_DIT     |

### Content of the Gain Table

For each distinct DIT value, this table logs:

| Name        | Description  |
|-------------|--|
| DIT         | Detector Integration Time in s   |
| MEAN_ON1    | Mean intensity of $on_1$ ( $< on_1 >$ )  |
| MEAN_ON2    | Mean intensity of $on_2$ ( $< on_2 >$ )  |
| MEAN_OFF1   | Mean intensity of $off_1$ ( $< off_1 >$ )  |
| MEAN_OFF2   | Mean intensity of $off_2$ ( $< off_2 >$ )  |
| SIG_ON_DIF  | Standard deviation in $on_1 - on_2$ ( $\sigma_{ondif}$ )                                 |
| SIG_OFF_DIF | Standard deviation in $off_1 - off_2$ ( $\sigma_{offdif}$ )                              |
| GAIN        | $\frac{(<on_1> + <on_2>) - (<off_1> + <off_2>)}{(\sigma_{ondif}^2 - \sigma_{offdif}^2)}$ |
| AUTOCORR    | Autocorrelation factor   |
| GAIN_CORR   | GAIN / AUTOCORR  |
| ADU         | $(<on_1> + <on_2>) - (<off_1> + <off_2>)/2$  |



|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 9 of 37                |

|                   |   |
|-------------------|---|
| X_FIT             | Independent (X) variable used in the fit  |
| X_FIT_CORR        | Independent (X) variable used in the fit (differs from X_FIT if autocorrelation active) |
| Y_FIT             | Dependent (Y) variable used in the fit  |
| Y_FIT_CORR        | Dependent (Y) variable used in the fit (differs from Y_FIT autocorrelation active)      |
| FLAG              | If 0 the corresponding frame is saturated   |
| ESO DET WINi UIT1 | user defined subintegration time  |

In case of saturated frames all columns but EXPTIME (or DIT) and FLAG are filled by -999.0.

The Coefficients Cube is created only if **pix2pix** is set to TRUE (default behaviour for this recipe). It contains as many planes as **order** + 1, each pixel in plane **i** containing the value of the coefficient or order **i** of the fitting computed for that position.

The Bad Pixel map depends on information contained in the Coefficients Cube, so it is created together with it.

The Bad Pixel Map contains only 0 for correct pixels and 1 for not linear ones. Non linear pixels are determined using the function `hdr1_bpm_fit_parameter_create_rel_chi` of HDRL which applies the chi-method: Relative cutoff on the chi distribution of all fits. Pixels with chi values which exceed of mean  $\pm$  kappa  $\pm$  standard deviation are considered bad.

The Difference and Autocorrelation images are only created if **intermediate** is set to TRUE (default behaviour for this recipe is FALSE). They represent intermediate steps of the Autocorrelation factor computation.

### 1.1.3 Quality control

**Lamp flux** QC LAMP FLUX in ADU/s.

**Detector gain** The computed gain, QC GAIN, and the corresponding error error, QC GAIN ERR, both in  $e^-$ /ADU units and its inverse QC CONAD in  $\text{ADU}/e^-$ , and the autocorrelation factor QC AUTCORFA.

And from the linearity part of the recipe:

**Lamp flux and stability** The lamp flux (QC LAMP FLUX) and its relative stability (QC LAMP STAB).

**Non linear coefficients** The polynomial fit coefficients (QC LIN COEFi, i=0-2) and the corresponding errors (QC LIN COEFi ERR).

**Effective non linearity** An additional QC parameter is QC LIN EFF, the effective non-linearity correction at a reference user defined level **ref\_level**:  $Q'(\text{ref\_level})$ .

**Number of non polynomial pixels** The recipe monitors the number of non linear pixels (QC NUM BPM).

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 10 of 37               |

**Fixed Pattern Noise** The detector Fixed Pattern Noise, QC FPN.

### 1.1.4 Parameters

```

--method          : Method to be used when computing GAIN. Methods
                   : appliable: <PTC | MED>. By default PTC method will be
                   : applied. [PTC]
--order           : Polynomial order for the fit (Linearity). [3]
--kappa           : Kappa value for the kappa-sigma clipping (Gain). [9.0]
--niter           : Number of iterations to compute rms (Gain). [5]
--llx            : x coordinate of the lower-left point of the region of
                   : interest. If not modified, default value will be 1.
                   : [-1]
--lly            : y coordinate of the lower-left point of the region of
                   : interest. If not modified, default value will be 1.
                   : [-1]
--urx            : x coordinate of the upper-right point of the region of
                   : interest. If not modified, default value will be X
                   : dimension of the input image. [-1]
--ury            : y coordinate of the upper-right point of the region of
                   : interest. If not modified, default value will be Y
                   : dimension of the input image. [-1]
--ref_level       : User reference level. [10000] ADU
--intermediate     : De-/Activate intermediate products. [FALSE]
--autocorr        : De-/Activate the autocorr option. [FALSE]
--collapse        : De-/Activate the collapse option. [FALSE]
--rescale         : De-/Activate the image rescale option. [TRUE]
--pix2pix         : De-/Activate the computation with pixel to pixel
                   : accuracy. [TRUE]
--bpmbin          : De-/Activate the binary bpm option. [TRUE]
--m               : Maximum x-shift for the autocorr. [26]
--filter          : Upper limit of Median flux to be filtered. [-1]
--n               : Maximum y-shift for the autocorr. [26]
--tolerance       : Tolerance for pair discrimination. [0.001]
--pafname         : Specific name for PAF file. [sinfo_rec_lingain]
--exts            : Activate the multi-exts option. [0]
--fpn_method      : Method for computing Fixed Pattern Noise (SMOOTH or
                   : HISTOGRAM). [HISTOGRAM]
--fpn_smooth      : template size in pixels for smoothing during FPN
                   : computation (only for SMOOTH method). [13]
--saturation_limit : all frames with mean saturation above the limit would
                   : not be used in calculation. [65535.0]
--gain_threshold  : all frames with mean value above the threshold would
                   : not be used in gain calculation. [65535.0]
--coeffs_cube_split : if TRUE, the recipe writes a COEFFS_CUBE product each

```

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 11 of 37               |

`extension in a separate file. [FALSE]`

## NOTE

Please note that to compute on all extensions the user must set the parameter **exts** to -1 (default is 0 - no extension case).

## 1.2 detmon\_opt\_lg

This recipe determines Linearity and Gain of detectors in the optical domain.

The Linearity reduction part computes the detector response as a function of the pixel intensity, fitting this function to a polynomial whose coefficients help to determine when it becomes non linear.

The Gain reduction part computes the Gain of the detector.

Both reduction steps are merged in this recipe as both accept the same input interface. There are some variants to both reduction parts, controlled by the input parameters. See (reference to Algorithms) for details.

This recipe determines the mean flux in five reference fields of the detector in order to measure contamination.

### 1.2.1 Input

This recipe expects input frames classified either as ON\_RAW (tag to be used to indicate FLAT exposures taken with the lamp on) or as OFF\_RAW (tag to be used to indicate bias frames). Each instrument pipeline adopting detmon algorithms may change this value. Please refer to each instrument corresponding recipe man page.

If no modification is done to the default behaviour, there must be two frames tagged ON\_RAW for each distinct EXPTIME value found among input frames.

The default behaviour for this recipe is having the parameter **collapse** is set to TRUE, which makes the recipe use the same pair of OFF\_RAW tagged frames for all EXPTIMES. The frames signal is integrate on a user defined region (**llx**, **lly**, **urx**, **ury**) before Linearity computation. If the parameter **collapse** is set to FALSE, the recipes expects the same kind of input as the detmon\_ir\_lg recipe, i.e. one pair of OFF\_RAW frames for each ON\_RAW pair.

The number of distinct EXPTIME values among input frames must be at least **order** + 1 as well.

### 1.2.2 Output

The following table lists all possible products of this recipe. Please note that not all of them are always created. Creation of some of these products depends on the value(s) of some input parameter(s).

If no modification is done to the default behaviour, this recipe produces only the first 2 products in the list.

Further details about conditions for product creation can be found along this section.

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 12 of 37               |

| Default recipe filename            | PRO.CATG     | Short description      |
|------------------------------------|--------------|------------------------|
| detmon_opt_lg_linearity_table.fits | DET_LIN_INFO | Linearity Table        |
| detmon_opt_lg_gain_table.fits      | GAIN_INFO    | Gain Table             |
| detmon_opt_lg_coeffs_cube.fits     | COEFFS_CUBE  | Coefficients cube      |
| detmon_opt_lg_bpm.fits             | BP_MAP_NL    | Bad pixel map          |
| detmon_opt_lg_diff_flat_i.fits     | DIFF_FLAT    | Difference images      |
| detmon_opt_lg_autocorr_i.fits      | AUTOCORR     | Autocorrelation images |

Both Linearity and Gain Tables are created in all possible execution cases. The last two products are generated only if **autocorr** is set to TRUE.

### Content of the Linearity Table

For each distinct EXPTIME value, this table logs:

| Name              | Description   |
|-------------------|---|
| EXPTIME           | Detector Exposure Time in s                               |
| MED               | Median intensity of $[(on_1 - off_1) + (on_2 - off_2)]/2$ |
| MEAN              | Mean intensity of $[(on_1 - off_1) + (on_2 - off_2)]/2$   |
| MED_DIT           | Ratio MED / DIT   |
| MEAN_DIT          | Ratio MEAN / DIT  |
| ADL               | Column DIT multiplied by mean value of column MED_DIT     |
| ESO DET WIN1 UIT1 | Corresponding raw frames FITS keyword values              |

### Content of the Gain Table

For each distinct EXPTIME value, this table logs:

| Name        | Description  |
|-------------|--|
| EXPTIME     | Detector Exposure Time in s  |
| MEAN_ON1    | Mean intensity of $on_1$ ( $< on_1 >$ )  |
| MEAN_ON2    | Mean intensity of $on_2$ ( $< on_2 >$ )  |
| MEAN_OFF1   | Mean intensity of $off_1$ ( $< off_1 >$ )  |
| MEAN_OFF2   | Mean intensity of $off_2$ ( $< off_2 >$ )  |
| SIG_ON_DIF  | Standard deviation in $on_1 - on_2$ ( $\sigma_{ondif}$ )                                 |
| SIG_OFF_DIF | Standard deviation in $off_1 - off_2$ ( $\sigma_{offdif}$ )                              |
| GAIN        | $\frac{(<on_1> + <on_2>) - (<off_1> + <off_2>)}{(\sigma_{ondif}^2 - \sigma_{offdif}^2)}$ |
| AUTOCORR    | Autocorrelation factor   |
| GAIN_CORR   | GAIN / AUTOCORR  |
| ADU         | $(< on_1 > + < on_2 >) - (< off_1 > + < off_2 >)/2$                                      |
| X_FIT       | Independent (X) variable used in the fit   |
| X_FIT_CORR  | Independent (X) variable used in the fit (differs from X_FIT if autocorrelation active)  |

|     |   |        |                        |
|-----|---|--------|------------------------|
| ESO | Detector monitoring recipes User Manual | Doc:   | VLT-MAN-ESO-19500-4846 |
|     |   | Issue: | Issue 1.3.12           |
|     |   | Date:  | Date 2023-03-14        |
|     |   | Page:  | 13 of 37               |

|                   |  |
|-------------------|--|
| Y_FIT             | Dependent (Y) variable used in the fit   |
| Y_FIT_CORR        | Dependent (Y) variable used in the fit (differs from Y_FIT autocorrelation active) |
| ESO DET WIN1 UIT1 | Corresponding raw frames FITS keyword values                                       |

The Coefficients Cube is created only if **pix2pix** is set to TRUE. It contains as many planes as **order** + 1, each pixel in plane **i** containing the value of the coefficient or order **i** of the fitting computed for that position.

The Bad Pixel map depends on information contained in the Coefficients Cube, so it is created together with it. Its content may vary depending on **bpm bin**.

The Bad Pixel Map contains only 0 for correct pixels and 1 for not linear ones. Non linear pixels are determined using the function `hdr1_bpm_fit_parameter_create_rel_chi` of HDRL which applies the chi-method: Relative cutoff on the chi distribution of all fits. Pixels with chi values which exceed of mean  $\pm$  kappa  $\pm$  standard deviation are considered bad.

The Difference and Autocorrelation images are only created if **intermediate** is set to TRUE (default behaviour for this recipe is FALSE). They represent intermediate steps of the Autocorrelation factor computation.

### 1.2.3 Quality control

**Lamp flux** QC LAMP FLUX in ADU/s.

**Detector gain** The computed gain, QC GAIN, and the corresponding Measured Square error, QC GAIN ERR, both in  $e^-$ /ADU units and its inverse QC CONAD in ADU/ $e^-$ , and the autocorrelation factor QC AUTCORFA.

**Fixed Pattern Noise** The detector Fixed Pattern Noise, QC FPN.

And from the linearity part of the recipe:

**Lamp flux RMS** The RMS of the lamp flux (QC LAMP STD).

**Non linear coefficients** The polynomial fit coefficients (QC LIN COEFi, i=0-2) and the corresponding errors (QC LIN COEFi ERR).

**Effective non linearity** An additional QC parameter is QC LIN EFF, the effective non-linearity correction at a reference user defined level **ref\_level**:  $Q'(\text{ref\_level})$ .

**Number of non polynomial pixels** The recipe monitors the number of non linear pixels (QC NUM BPM).

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 14 of <a href="#">37</a> |

## 1.2.4 Parameters

```

--method          : Method to be used when computing GAIN. Methods
                   : applicable: <PTC | MED>. By default PTC method will be
                   : applied. [PTC]
--order           : Polynomial order for the fit (Linearity). [3]
--kappa           : Kappa value for the kappa-sigma clipping (Gain). [3]
--niter           : Number of iterations to compute rms (Gain). [5]
--llx             : x coordinate of the lower-left point of the region of
                   : interest. If not modified, default value will be 1.
                   : [-1]
--lly             : y coordinate of the lower-left point of the region of
                   : interest. If not modified, default value will be 1.
                   : [-1]
--urx             : x coordinate of the upper-right point of the region of
                   : interest. If not modified, default value will be X
                   : dimension of the input image. [-1]
--ury             : y coordinate of the upper-right point of the region of
                   : interest. If not modified, default value will be Y
                   : dimension of the input image. [-1]
--ref_level       : User reference level. [10000]
--intermediate    : De-/Activate intermediate products. [FALSE]
--autocorr        : De-/Activate the autocorr option. [FALSE]
--collapse        : De-/Activate the collapse option. [TRUE]
--rescale         : De-/Activate the image pair rescaling option. [TRUE]
--pix2pix         : De-/Activate the computation with pixel to
                   : pixel accuracy. [FALSE]
--bpmbin          : De-/Activate the binary bpm option. [FALSE]
--m               : Maximum x-shift for the autocorr. [26]
--filter          : Upper limit of Median flux to be filtered. [-1]
--n               : Maximum y-shift for the autocorr. [26]
--tolerance       : Tolerance for pair discrimination. [0.001]
--pafname         : Specific name for PAF file. [detmon_opt_lg]
--exts            : Activate the multi-exts option. [0]
--llx1            : x coord of the lower-left point of the first field
                   : used for contamination measurement. If not modified,
                   : default value will be 1. [-1]
--lly1            : y coord of the lower-left point of the first field
                   : used for contamination measurement. If not modified,
                   : default value will be 1. [-1]
--urx1            : x coord of the upper-right point of the first field
                   : used for contamination measurement. If not modified,
                   : default value will be X dimension of the input image.
                   : [-1]
--ury1            : y coord of the upper-right point of the first field

```

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 15 of 37               |

used for contamination measurement. If not modified, default value will be Y dimension of the input image. [-1]

--llx2 : x coord of the lower-left point of the second field used for contamination measurement. If not modified, default value will be 1. [-1]

--lly2 : y coord of the lower-left point of the second field used for contamination measurement. If not modified, default value will be 1. [-1]

--urx2 : x coord of the upper-right point of the second field used for contamination measurement. If not modified, default value will be half of the X dimension of the input image. [-1]

--ury2 : y coord of the upper-right point of the second field used for contamination measurement. If not modified, default value will be half of the Y dimension of the input image. [-1]

--llx3 : x coord of the lower-left point of the third field used for contamination measurement. If not modified, default value will be 1. [-1]

--lly3 : y coord of the lower-left point of the third field used for contamination measurement. If not modified, default value will be half of the Y dimension of the input image. [-1]

--urx3 : x coord of the upper-right point of the third field used for contamination measurement. If not modified, default value will be half of X dimension of the image. [-1]

--ury3 : y coord of the upper-right point of the third field used for contamination measurement. If not modified, default value will be Y dimension of the image. [-1]

--llx4 : x coord of the lower-left point of the fourth field used for contamination measurement. If not modified, default value will be half of X dimension of the image. [-1]

--lly4 : y coord of the lower-left point of the fourth field used for contamination measurement. If not modified, default value will be half of the Y dimension of the input image. [-1]

--urx4 : x coord of the upper-right point of the fourth field used for contamination measurement. If not modified, default value will be X dimension of the image. [-1]

--ury4 : y coord of the upper-right point of the fourth field used for contamination measurement. If not modified, default value will be Y dimension of the input image.

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 16 of 37               |

```

                                [-1]
--llx5                        : x coord of the lower-left point of the fifth field
                                used for contamination measurement. If not modified,
                                default value will be half of the X dimension of the
                                input image. [-1]
--lly5                        : y coord of the lower-left point of the fifth field
                                used for contamination measurement. If not modified,
                                default value will be 1. [-1]
--urx5                        : x coord of the upper-right point of the fifth field
                                used for contamination measurement. If not modified,
                                default value will be X dimension of the image. [-1]
--ury5                        : y coord of the upper-right point of the fifth field
                                used for contamination measurement. If not modified,
                                default value will be half of Y dimension of the
                                input image. [-1]
--saturation_limit           : all frames with mean saturation above the limit would
                                not be used in linearity calculation. [65535.0]
--gain_threshold             : all frames with mean value above the threshold would
                                not be used in gain calculation. [65535.0]

```

### 1.3 Multi-region recipes

The recipes `detmon_opt_lg_mr` and `detmon_ir_lg_mr` are wrappers around the `detmon_opt_lg` and `detmon_ir_lg` recipes to handle multiple subregions of a detector in a single invocation.

They are functionally equivalent to calling their parent recipes multiple times with different settings for the region of interest (`--llx` etc.), except that the output produced is merged into single files.

These recipes currently do not support specifying contamination regions and the `--intermediate` option.

#### 1.3.1 Input

The input for the multi-region recipes is the same as for `detmon_opt_lg` and `detmon_ir_lg`. The only difference is that the region of interest is specified with the `--regions` parameter which must be formatted as a colon separated list of regions with four points each, comma separated. The ordering of points is `llx,lly,urx,ury`. For example two regions from pixel 1 to 20 and pixel 21 to 40:

```
--regions=1,1,20,20:21,21,40,40
```

Note that the coordinates are according to the FITS convention beginning from 1 to the upper value which is inclusive.

#### 1.3.2 Output

The output is structurally similar to the output of the single region variants, only the differences are listed here.



|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 17 of <a href="#">37</a> |

All FITS header keys have `REGi` prepended to their name. The `i` is an integer defining the region as ordered in the `--regions` parameter of the recipe. Additionally it contains the `REGi {LLX, LLY, URX, URY}` keys defining the region coordinates that were provided and used in the computations.

The output images are the same as in the single region recipes except they only contain values in the specified regions and everything else is set to zero.

The tables contain an additional integer column defining the region the row belongs to. The integers are the same as used for the FITS header keys and are defined by the ordering of the regions on the command line.

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 18 of <a href="#">37</a> |

## 2 Algorithms

In this section the main algorithms implemented within the Detector Monitoring Project are described.

Relevant data reduction parameters are typed in **bold** face. For convenience we omit the common prefix **detmon.** for the recipe description as well as the step prefix name for the algorithm description.

### 2.1 Basic Algorithms

#### 2.1.1 Histogram computation

This algorithm build an histogram of the input frame intensity values. Input parameters are the minimum and maximum allowed frame intensity values, **histo\_min**, **histo\_max** in order to remove the contribution of threshold pixels, and the number of histogram sampling points **histo\_npoints**. The interval [**histo\_min**, **histo\_max**] is divided in equal steps of size  $\frac{\text{histo\_max} - \text{histo\_min}}{\text{histo\_npoints}}$ , and an histogram of the frame intensity values is build. The histogram maximum and its corresponding bin is determined (this is the frame intensity clean mean). Then a Gaussian is fit to the histogram. The Gaussian sigma gives the frame intensity clean sigma. To make the fit more robust one may fit the logarithm of the intensity as a function of the square of the histogram frequency (abscissa). This is a linear function and the corresponding fit should be quite robust and reliable.

#### 2.1.2 Iteratives clean mean and sigma computation

This algorithm computes a clean mean and standard deviation through iterative kappa-sigma clipping of outliers. At each iteration the value of the clean mean and sigma is updated. Optionally the user may set the minimum and maximum allowed intensity values (**threshold\_min**, **threshold\_max**). Initially the input frame intensity values outside the interval [**threshold\_min**, **threshold\_max**] are flagged to remove their contribution from the following operations. Then an initial value of the intensity mean and stdev are computed. A loop is repeated **niter** iterations, and in each iteration are clipped intensity values which are outside the interval [**mean-kappa\*stdev**, **mean+kappa\*stdev**]. The final values of mean and stdev are the clean frame intensity mean and sigma.

#### 2.1.3 Detector contamination computation: **detmon\_nir\_contamination**, NIR domain

This recipe measures the median flux in five fields of a reduced flat field. Input are a set of pairs of on/off frames. For each pair the off frame is subtracted from the on frame. Then the median flux of 5 user definable reference regions (**ilxi**, **llyi**, **urxi**, **uryi**,  $i=1-5$ ) is computed. From the computed values of each reference region one takes the median and defines the following QC parameters: QC CONTAM<sub>i</sub> ( $i=1-5$ ).

### 2.2 General Algorithms

There are 2 different methods to compute the Gain, Photon Transfer Curve (PTC) and Median. We explain them here:

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 19 of 37               |

### 2.2.1 PTC and MED Algorithms for Gain computation

To measure the detector's gain, a single algorithm is implemented common to both optical and infrared wavelength domains. In optical and infrared domain, the input is different. In the following description what we indicate as *off* frame is in the near infrared an off-lamp frame, but in the optical domain is a bias frame.

Input frames for NIR are a set of pairs of on/off flats, each set with a different DIT.

If we indicate with  $on_{dif}$  the on frames difference:

$$on_{dif} = on_2 - on_1 \quad (1)$$

If we indicate with  $of_{dif}$  the off frames difference:

$$of_{dif} = of_2 - of_1 \quad (2)$$

If we indicate with  $mon_i$  the on frame clean mean (obtained either with the algorithm 2.1.1 or with the algorithm 2.1.2) relative to the integration time (EPTIME or DIT)  $i$ :

$$mon_i = mean(on_i) \quad (3)$$

And with  $mof_i$  the similar off frames clean mean:

$$mof_i = mean(of_i) \quad (4)$$

One may find the gain by fitting the following linear (one may even fit a parabola and verify that the coefficient of the quadratic term is small enough, this being a QC parameter) relation:

$$(mon_1 + mon_2) - (mof_1 + mof_2)_i = NDIT \times gain(e^-/ADU)(\sigma_{ondif}^2 - \sigma_{ofdif}^2)_i \alpha \quad (5)$$

where  $\alpha$  is the autocorrelation function computed as:

$$\alpha = \sum_{m,n} R_{m,n} \quad (6)$$

where:

$$R_{m,n} = \frac{\sum_{i,j} V_{i,j} * V_{i+m,j+n}}{\sum_{i,j} V_{i,j}^2} \quad (7)$$

where  $V_{i,j}$  is the intensity of the pixel  $(i, j)$ .

The values of  $mon_1$ ,  $mof_2$ ,  $mof_1$ ,  $mof_2$ ,  $\sigma_{ondif}^1$ ,  $\sigma_{ondif}^2$  from equation 5 together with the values of the frame exposure time (DIT or EXPTIME) for each frame set should be logged in the corresponding columns of the product table having PRO.CATG set to GAIN\_INFO.

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 20 of 37               |

If the parameter **method** is set to PTC (Photon Transfer Curve method) the gain is obtained with a linear fit of the values in eq. 5. Else, if the **method** is set to MED, the gain is obtained computing a simple median of the gain values that can be obtained with equation 5.

$$conad = gain^{-1} \quad (8)$$

The gain and its error are computed and logged as QC parameters (QC GAIN, QC GAIN MSE, in  $e^-/ADU$  and QC CONAD in  $ADU/e^-$ , QC AUTCORFA).

In eq. 5 it is critical to compute a clean mean and a clean standard deviation. Two methods are possible:

1. Build a histogram and compute the histogram centre and sigma (2.1.1).
2. Determine the mean and standard deviation repeating **niter** a clean mean computation with kappa-sigma cleaning of bad pixels (2.1.2).

**Saturation Limit** Frames with saturation level above the limit, defined by the **saturation\_limit** parameter value, and all the following frames (with exposure time equal or greater than the one of the first saturated frame) are not considered into the calculation. The same rule is used for the calculation of the detector linearity.

**Empty frames** In some cases, an extension could contain 0 values for all pixels and the gain can not be computed. For example it in the windowing mode of CRIRES the 1st and 4th detectors have all pixels values set to 0. Such event is caught and the gain, the linearity coefficients and the FPN are not computed. The recipes (**detmon\_opt\_lg** and **detmon\_ir\_lg**) produce the following warning:

```
[WARNING] detmon_ir_lg:
  An empty frame has been detected, linearity, coeffs, gain, FPN values will not
```

The output products for the "empty" extension will:

- miss QC parameters GAIN and GAIN\_ERR;
- have QC.LINEFF set to 0;
- have the gain table with 0 values;
- have the COEFFS\_CUBE product containing 0 values for the "empty" extensions;
- have the BP\_MAP\_NL product (bad pixels map) with all pixels are marked as "bad".

### 2.2.2 Detector linearity computation (IR): irplib\_detmon\_nir\_linear

The input interface is the same to irplib\_detmon\_gain, that is, for each exposure time value two pairs of frames (a pair ON and a pair OFF) are provided. The input frames DPR keyword should be (we recommend the following, but each instrument pipeline may decide to adopt also different values):

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 21 of 37               |

| DPR CATG | DPR TECH | DPR TYPE              |
|----------|----------|-----------------------|
| CALIB    | IMAGE    | LINEARITY,FLAT,DETCAR |

For each exposure time value, two difference frames are computed:  $dif_1 = on_1 - of_1$  and  $dif_2 = on_2 - of_2$ . These two difference frames are then averaged pixel by pixel. Then the mean and the median values of the difference frame in a rectangle defined by **llx**, **lly**, **urx** and **ury**, are computed and written in the output table (PRO.CATG=LIN\_INFO), in the column MEAN and MED, together with the corresponding pair integration time (DIT or EXPTIME), and the corresponding ratio, MEAN\_DIT and MEAN\_DIT.

Each average frame corresponding to a given DIT value is stored in a different plane of cube. Then for each x,y pixel intensity I(x,y,t) is computed a polynomial fit of order **order** in direction t.

$$Q(x, y, t) = \sum_{0, \text{order}} c_i(x, y) t^i \quad (9)$$

$$Q'(x, y, t) = \frac{Q(x, y, t)}{c_1(x, y) t} \quad (10)$$

The corresponding coefficients  $c_i$  are stored in a cube of size **order**+1. This is a recipe product (LIN\_COEF). In principle the coefficients in the constant-term plane  $c_0$  should be zero. The constant-term in the polynomial fit is nevertheless still fitted. This provides an option for verifying that the subtraction of the ON and OFF data works as intended.

Non linear pixels are flagged using the High level Data Reduction Library function implementing the bad pixel detection method4 (pval) on a sequence of images. This method uses the p-value (between 0% and 100%) to discriminate between good and bad pixels. Fits with a p-value below the threshold are considered to be bad pixels. Recipe testing has shown that non linear pixels are stably detected using a (recipe hard-coded) pval value of 0.001.

Finally the median and standard deviation of each coefficient plane are monitored as quality control parameters (QC LIN COEFi, QC LIN COEFi ERR).

$$Q(t) = \sum_{0, \text{order}} c_i t^i \quad (11)$$

$$Q'(t) = \frac{Q(t)}{c_1 t} \quad (12)$$

Alternatively the detector linearity may be monitored on the frame obtained by integrating the signal over a user defined (**llx**, **lly**, **urx**, **ury**) detector region.

For each input frame on-off pair the difference frame is computed:  $dif = on - of$ . Each frame difference is stored on an imagelist. Then the clean mean/median and standard deviation are computed on a given frame region (**llx**, **lly**, **urx**, **ury**). Those values are computed for each frame difference and stored in a vector. Then computed a polynomial fit of order **order** of vector values is computed:

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 22 of 37               |

$$P(t) = \sum_{0, \text{order}} a_i t^i \quad (13)$$

$$P'(t) = \frac{P(t)}{a_1 t} \quad (14)$$

The corresponding coefficients  $a_i$  are monitored as quality control parameters (QC LIN COEFi, QC LIN COEFi ERR).

Note that  $Q'(t)$  (or  $P'(t)$ ) is independent from the lamp flux.

The correction that can be applied in a later step of the cascade would be:

$$I_{corrected} = I_{original}/Q'(t) \quad (15)$$

A map of non linear pixels is also generated.

An additional QC parameter is QC LIN EFF, the effective non-linearity correction at a reference user defined level **ref\_level**:  $Q'(\text{ref\_level})$ .

### 2.2.3 Detector linearity computation (optical): irplib\_detmon\_opt\_linear

**Pixel to pixel linearity computation** The algorithm is almost identical to irplib\_detmon\_nir\_linear, with the difference that usually one does not have as many biases as flat fields. In that case in computing the differences  $diff_1 = on_1 - off_1$  and  $diff_2 = on_2 - off_2$ , the *off* frames should be considered either always the same two raw biases in the input list (which set as requirement to have at least two raw biases), or the same master bias (which allows to have in input any number of raw biases or even a reference master bias frame, in which case  $off_2 = off_1 = MBIAS$ ).

**Mean linearity computation** In this case (or more generally if the **collapse** option is activated as by default in optical domain) the linearity is computed in a similar way to the NIR (pixel to pixel), but a master bias frame is used in place of an off frame and instead of performing a pixel to pixel computation the signal is first integrated over the pixels within a given rectangle.

### 2.2.4 Fixed Pattern Noise determination

The linearity recipe computes also the detector fixed pattern noise.

At first it is selected the input not saturated flat frame with the maximum flux. To make sure that a frame is not saturated the user can specify a threshold value with the input parameter **thresh**. Then the recipe does not consider all the frames having flux greater than the threshold or exposure time greater of the first frame having flux above the threshold. This second condition is required to make sure that “super-saturated” frames (frames saturated but with flux below the threshold) are considered.

Then it is computed the robust noise over a given region (**ref\_llx,ref\_lly, ref\_urx, ref\_ury**)

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 23 of 37               |

This operation can be performed using two possible methods:

1. HISTOGRAM. An histogram over a user specified region is computed and a Gaussian fit is performed. The sigma of the fit is considered, be this  $F_{tot}$ .
2. SMOOTH. High spatial frequency variations in the signal are smoothed by a squared filter of size **fpn\_smooth**  $\times$  **fpn\_smooth**, where **fpn\_smooth** is a recipe input parameter. Then the smoothed image is normalized by the median computed over the user defined region, and then an histogram method is applied to the normalized image. This results again into a  $F_{tot}$  value.

This is smoothed with a squared filter of size **fpn\_smooth**  $\times$  **fpn\_smooth**, where **fpn\_smooth** is a recipe input parameter, and then normalized by the median of the flat frame. Finally the fixed pattern noise  $fpn$  is given by the following formula:

$$fpn[ADU] = \sqrt{F_{tot}^2 - \frac{FA}{gain}}$$

where  $FA$  is the median flux value over the same frame. and gain is the detector gain value previously computed by the same recipe.

Finally the recipe compute the ratio:

$$fpnr = \frac{fpn}{FA}$$

This last value is log as QC parameters (QC.FPN).

## 2.3 Recipe Algorithms

### 2.3.1 detmon\_ir\_lg, detmon\_opt\_lg

This section describes the algorithm used for Linearity and Gain computation both in the IR and in the optical domain. Both are described here together, as they are, in fact, the same algorithm with different default values. Additions specific to the optical domain will be detailed at the end of this section.

- After an initial check on the input data, the recipe parameters, the number of extensions in the input frames is determined.
- The recipe looks for frames tagged ON and frames tagged OFF in the input set of frames file. Frames with no tag or unrecognized tag are simply ignored. In the optical case ON corresponds to FLAT frames and OFF, to BIAS frames. The boolean parameter **collapse** is by default inactive in the IR case and active in the optical one.

Then the ON (and in IR, as **collapse**=FALSE, also the OFF frames) are ordered by increasing DIT.

- Then for each input frame extension the following operations are performed:

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 24 of <a href="#">37</a> |

- A table to hold results of the detector's gain computation is created with the following columns: DIT (in NIR domain) or EXPTIME (in optical domain), MEAN\_ON1, MEAN\_ON2, MEAN\_OFF1, MEAN\_OFF2, SIG\_ON\_DIF, SIG\_OFF\_DIF, GAIN, AUTOCORR, GAIN\_CORR, ADU, X\_FIT, X\_FIT\_CORR, Y\_FIT.
- A table to hold results of the detector's linearity computation is created with the following columns: DIT (in NIR domain) or EXPTIME (in optical domain), MED, MEAN, MED\_DIT, MEAN\_DIT, ADL.
- If **collapse**=FALSE (default value for NIR domain) a lamp stability check is performed. The recipe:
  - controls that ON and OFF frames are taken with the same (within a given threshold, set to 0.001 s) integration time (DIT in NIR and EXPTIME in optical domain).
  - verifies that frames used to verify the lamp stability are not consecutive.
  - verifies that there are enough (at least 3) frames to perform a stability check
  - For each frame pair determines the median of the frame difference ON - OFF.
  - then determines the maximum relative difference to the first pair. This is an output QC parameter of the recipe (ESO.QC.LAMP.STAB).
  - Finally verifies that this maximum is less then 0.01.
- Then a frame pair of same integration time is extracted.  
 In the IR case (**collapse** inactive) we seek for pairs (sets of 2 frames having equal exposure time) in both ON and OFF sub-framesets. The recipe requires in this case each ON pair to have a matching OFF pair.  
 In the optical case (**collapse** active) we seek only for ON pairs. There must be an OFF sub-frameset, but its characteristical integration time doesn't need to match the one of the ON frame.
- if(**filter** > 0) the median of each pair on a a region defined by parameters **llx**, **lly**, **urx**, **ury** is determined. If one of the two computed medians exceeds a given threshold specified by the parameter **filter**, the given pair is not considered in the following data reduction.  
 Then the recipe uses the parameter **saturation\_limit** to select out from the linearity analysis frames whose robust (kappa-sigma clipped) mean over the user defined [**llx**,**lly**,**urx**,**ury**] region is greater than the **saturation\_limit** value.  
 Moreover the recipe uses additionally the **gain\_threshold** parameter in order to select out frames whose robust (kappa-sigma clipped) mean over the user defined [**llx**,**lly**,**urx**,**ury**] region is greater than the **gain\_threshold** value.
- if(**filter** < 0)
  - if **collapse**=TRUE a pair of off frames is extracted from the corresponding frame set. Else, if **collapse**=FALSE, (default case for optical domain), the previously computed master bias is considered as reference off frame.
  - If the option **rescale** is active (**rescale**=TRUE), each first frame of a pair is rescaled to the median value of the second one, to reduce illumination differences due to possible residual lamp instability. This is particularly efficient (and necessary) when autocorrelation factor (option **autocorr** is taken into account for Gain computation. Once all pairs have been tagged (and rescaled, if required),
  - The detector linearity is measured (see [2.2.2](#) or [2.2.3](#)).



|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 25 of <a href="#">37</a> |

- In optical domain the detector contamination is determined.
- Finally the detector gain is measured. There are two possible procedures for Gain computation (depending on the option **method**, see [2.2.1](#) and [2.2.1](#)) and two for Linearity computation (depending on the option **pix2pix**, see [2.2.2](#) and [2.2.3](#)).

### Count rate

From the first frame set the lamp count rate is computed as:

$$cr = \frac{mean(on_1 - off_1)}{exptime_1} \quad (16)$$

monitored as a quality control parameter (QC LAMP FLUX in ADU/s).

Note:

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 26 of <a href="#">37</a> |

### 3 Usage of common detlin recipe across instruments

Here we give brief information how to execute a common recipe (in our case `detmon_opt_lg` or `detmon_ir_lg`) to reduce data from different instruments. More information on EsoRex can be found on [3] and the relevant links. The main EsoRex features can be obtained by typing on command line:

```
esorex --help
```

We do not recommend to use the environment variables (as indicated by the `esorex` on line help) to set the possible parameters but instead to use the appropriate `esorex` command line options.

#### 3.1 How to execute an instrument specific and detmon pipelines

The user has to:

1. To customize EsoRex behavior, the user can possibly by edit the file `$HOME.esorex.esorex.rc` that can be generated by typing  

```
esorex --create-config
```
2. For example the user may indicate the location of the instrument pipeline recipes by setting a proper value to the variable  

```
esorex.caller.recipe-dir
```

The EsoRex configuration file may also be specified on command line by setting the `esorex` command line option `--config`
3. The user may want to customize a given instrument pipeline behavior with a specific configuration file. This setting may be part of the Association Block (AB).  

This can be done by using the `--recipe-config esorex` parameter option.
4. The user may eventually pass on the command line a given parameter list with appropriate values.

#### 3.2 Handling parameter setting priorities

The `esorex` and recipes parameters are prioritized as follows:

1. `esorex` has some default behavior.
2. The file `$HOME.esorex.esorex.rc` may be edited to customize the `esorex` default behavior.
3. The file passed as `esorex --config` parameter has precedence on previous settings.
4. Each recipe has a default behavior.
5. The file `$HOME.esorexrecipe.rc` may be used to customize the recipe behavior.

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 27 of <a href="#">37</a> |

6. The file passed as esorex --recipe-config parameter value has precedence of the previous recipe parameter settings.
7. Any additional parameter passed on command line has precedence on previous settings.

### 3.2.1 How to activate instrument specific and detmon pipelines

The command

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

generates the configuration file

```
$HOME/.esorex/esorex.rc
```

that specifies relevant esorex configuration parameters. In order to activate instrument specific and detmon pipelines the user has to set the variable esorex.caller.recipe.dir to:

```
esorex.caller.recipe.dir=/home/quality/pipelines/{pipeline_instrume_version}:  
/home/quality/pipelines/detmon_{version}
```

where pipeline\_instrume\_version indicates a given instrument specific pipeline release and detmon\_{version} indicates a a given detmon pipeline release. For example

```
esorex.caller.recipe.dir=/home/quality/pipelines/fors2-4.6.6:  
/home/quality/pipelines/detmon-1.0.0
```

Is a possible setting.

The user can verify at the command line what recipes are available by typing:

```
esorex --recipes
```

### 3.2.2 How to de-activate detmon pipelines

To deactivate detmon recipes it is sufficient to remove the corresponding installation path location from the esorex.caller.recipe.dir setting:

```
esorex.caller.recipe.dir=/home/quality/pipelines/{pipeline\_instrume\_version}
```

For example:

```
esorex.caller.recipe.dir=/home/quality/pipelines/fors2-4.6.6
```

|     |   |        |                        |
|-----|---|--------|------------------------|
| ESO | Detector monitoring recipes User Manual | Doc:   | VLT-MAN-ESO-19500-4846 |
|     |   | Issue: | Issue 1.3.12           |
|     |   | Date:  | Date 2023-03-14        |
|     |   | Page:  | 28 of 37               |

### 3.2.3 How to specify instrument-specific input parameters

A recipe may be tailored to a given instrument by setting the relevant parameter in an instrument configuration file and then passing this on the esorex command line, like:

`esorex --recipe-config=detmon_opt_fors2.rc detmon_opt_lg fors_detmon_opt_lg.sof`

where the file `fors_detmon_opt_fors2.rc` will look like:

```
# File: /home/amodigli/.esorex/detmon_opt_lg.rc
#
# Note: This configuration file has been automatically
#       generated by the esorex (v3.8.3) program.
#
# Date: 14-Oct-2010 10:52:40
#
#
# --method
# Method to be used when computing GAIN. Methods applicable: <PTC | MED>. By
# default PTC method will be applied.
detmon.detmon_opt_lg.method=PTC
#
# --order
# Polynomial order for the fit (Linearity).
detmon.detmon_opt_lg.order=2
#
# --kappa
# Kappa value for the kappa-sigma clipping (Gain).
detmon.detmon_opt_lg.kappa=4.0
#
# --niter
# Number of iterations to compute rms (Gain).
detmon.detmon_opt_lg.niter=5
#
# --llx
# x coordinate of the lower-left point of the region of interest. If not
# modified, default value will be 1.
detmon.detmon_opt_lg.llx=1
#
# --lly
# y coordinate of the lower-left point of the region of interest. If not
# modified, default value will be 1.
detmon.detmon_opt_lg.lly=6
#
# --urx
# x coordinate of the upper-right point of the region of interest. If not
# modified, default value will be X dimension of the input image.
detmon.detmon_opt_lg.urx=2048
#
# --ury
# y coordinate of the upper-right point of the region of interest. If not
# modified, default value will be Y dimension of the input image.
detmon.detmon_opt_lg.ury=1029
#
# --ref_level
# User reference level.
detmon.detmon_opt_lg.ref_level=10000
#
# --intermediate
# De-/Activate intermediate products.
detmon.detmon_opt_lg.intermediate=FALSE
```

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 29 of 37               |

```

# --autocorr
# De-/Activate the autocorr option.
detmon.detmon_opt_lg.autocorr=FALSE

# --collapse
# De-/Activate the collapse option.
detmon.detmon_opt_lg.collapse=TRUE

# --rescale
# De-/Activate the image rescale option.
detmon.detmon_opt_lg.rescale=TRUE

# --pix2pix
# De-/Activate the computation with pixel to pixel accuracy.
detmon.detmon_opt_lg.pix2pix=TRUE

# --bpmbin
# De-/Activate the binary bpm option.
detmon.detmon_opt_lg.bpmbin=FALSE

# --m
# Maximum x-shift for the autocorr.
detmon.detmon_opt_lg.m=26

# --filter
# Upper limit of Median flux to be filtered.
detmon.detmon_opt_lg.filter=-1

# --n
# Maximum y-shift for the autocorr.
detmon.detmon_opt_lg.n=26

# --tolerance
# Tolerance for pair discrimination.
detmon.detmon_opt_lg.tolerance=0.1

# --pafname
# Specific name for PAF file.
detmon.detmon_opt_lg.pafname=detmon_opt_lg

# --exts
# Activate the multi-exts option.
detmon.detmon_opt_lg.exts=0

# --fpn_method
# Method for computing Fixed Pattern Noise (SMOOTH or HISTOGRAM).
detmon.detmon_opt_lg.fpn_method=HISTOGRAM

# --fpn_smooth
# template size in pixels for smoothing during FPN computation (only for SMOOTH
# method).
detmon.detmon_opt_lg.fpn_smooth=13

# --saturation_limit
# all frames with mean saturation above the limit would not be used in
# linearity calculation.
detmon.detmon_opt_lg.saturation_limit=65535.0

# --gain_threshold
# all frames with mean value above the threshold would not be used in
# gain calculation.
detmon.detmon_opt_lg.gain_threshold=65535.0

# --llx1
# x coord of the lower-left point of the first field used for contamination

```

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 30 of 37               |

```
# measurement. If not modified, default value will be 1.
detmon.detmon_opt_lg.llx1=-1

# --lly1
# y coord of the lower-left point of the first field used for contamination
# measurement. If not modified, default value will be 1.
detmon.detmon_opt_lg.lly1=-1

# --urx1
# x coord of the upper-right point of the first field used for contamination
# measurement. If not modified, default value will be X dimension of the
# input image.
detmon.detmon_opt_lg.urx1=-1

# --ury1
# y coord of the upper-right point of the first field used for contamination
# measurement. If not modified, default value will be Y dimension of the
# input image.
detmon.detmon_opt_lg.ury1=-1

# --llx2
# x coord of the lower-left point of the second field used for contamination
# measurement. If not modified, default value will be 1.
detmon.detmon_opt_lg.llx2=-1

# --lly2
# y coord of the lower-left point of the second field used for contamination
# measurement. If not modified, default value will be 1.
detmon.detmon_opt_lg.lly2=-1

# --urx2
# x coord of the upper-right point of the second field used for contamination
# measurement. If not modified, default value will be half of the X dimension
# of the input image.
detmon.detmon_opt_lg.urx2=-1

# --ury2
# y coord of the upper-right point of the second field used for contamination
# measurement. If not modified, default value will be half of the Y dimension
# of the input image.
detmon.detmon_opt_lg.ury2=-1

# --llx3
# x coord of the lower-left point of the third field used for contamination
# measurement. If not modified, default value will be 1.
detmon.detmon_opt_lg.llx3=-1

# --lly3
# y coord of the lower-left point of the third field used for contamination
# measurement. If not modified, default value will be half of the Y dimension
# of the input image.
detmon.detmon_opt_lg.lly3=-1

# --urx3
# x coord of the upper-right point of the third field used for contamination
# measurement. If not modified, default value will be half of X dimension of
# the image.
detmon.detmon_opt_lg.urx3=-1

# --ury3
# y coord of the upper-right point of the third field used for contamination
# measurement. If not modified, default value will be Y dimension of the
# image.
detmon.detmon_opt_lg.ury3=-1
```

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 31 of 37               |

```
# --llx4
# x coord of the lower-left point of the fourth field used for contamination
# measurement. If not modified, default value will be half of X dimension of
# the image.
detmon.detmon_opt_lg.llx4=-1

# --lly4
# y coord of the lower-left point of the fourth field used for contamination
# measurement. If not modified, default value will be half of the Y dimension
# of the input image.
detmon.detmon_opt_lg.lly4=-1

# --urx4
# x coord of the upper-right point of the fourth field used for contamination
# measurement. If not modified, default value will be X dimension of the
# image.
detmon.detmon_opt_lg.urx4=-1

# --ury4
# y coord of the upper-right point of the fourth field used for contamination
# measurement. If not modified, default value will be Y dimension of the
# input image.
detmon.detmon_opt_lg.ury4=-1

# --llx5
# x coord of the lower-left point of the fifth field used for contamination
# measurement. If not modified, default value will be half of the X dimension
# of the input image.
detmon.detmon_opt_lg.llx5=-1

# --lly5
# y coord of the lower-left point of the fifth field used for contamination
# measurement. If not modified, default value will be 1.
detmon.detmon_opt_lg.lly5=-1

# --urx5
# x coord of the upper-right point of the fifth field used for contamination
# measurement. If not modified, default value will be X dimension of the
# image.
detmon.detmon_opt_lg.urx5=-1

# --ury5
# y coord of the upper-right point of the fifth field used for contamination
# measurement. If not modified, default value will be half of Y dimension of
# the input image.
detmon.detmon_opt_lg.ury5=-1

#
# End of file
```

In case of NIR data the syntax is similar. As far as we know, QC-Garching reduces pipeline data by generating and processing Association Blocks (ABs). Those ABs define several things, including the actual command to be executed. We recommend to modify that command by specifying, in addition to the actual instrument specific recipe name, eventual parameter list, and set of frames, also a parameter setting like:

```
--recipe-config=detmon_opt_fors2.rc
```

Note that a first version of the detmon\_opt\_fors2.rc can be delivered by the pipeline developer but need to be maintained either by the QC scientist or in agreement with the QC scientist, who is the best person to verify

|            |  |        |                        |
|------------|--|--------|------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846 |
|            |  | Issue: | Issue 1.3.12           |
|            |  | Date:  | Date 2023-03-14        |
|            |  | Page:  | 32 of 37               |

and eventually improve the values of the parameters that are specified in that configuration file, depending on the instrument operations needs and history. An up to date list of instrument specific rc files can be found on: <http://www.eso.org/qc/docu/detmon/index.html>

### 3.2.4 Processing linearity data with multi extensions

The linearity recipe provides one parameter, **exts**, to control the recipe behavior with data having a different number of extensions. The parameter default value is -1. In this case the recipe process all extensions present and generates products having a number of extensions equal to the one of the raw data. If the user would like to process a given extension (n), the value n (n=0,1,...N; N= total number of extensions), should be used. In this case the recipe will generate products with only one extension containing results corresponding to the extension indicated on command line.

### 3.2.5 Instrument specific configuration files

Current lingain recipes implementation have been verified with several operational PLS pipelines. Some of them require some parameter to be different from the default. We highlight below those:

#### UT1

- CRIRES

```
llx=1
lly=15
urx=1024
ury=345
ref_level=6000
method=PTC
exts=-1
```

- FORS2 (parameter values provided by ESO QC scientist)

```
FORS2_chip2_2x2
"--llx=202";
"--lly=323";
"--urx=1854";
"--ury=1027";
"--llx1=928";
"--lly1=575";
"--urx1=1128";
"--ury1=775";
"--llx2=202";
"--lly2=323";
```



|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 33 of <a href="#">37</a> |

```

"--urx2=402";
"--ury2=523";
"--llx3=202";
"--lly3=827";
"--urx3=402";
"--ury3=1027";
"--llx4=1654";
"--lly4=323";
"--urx4=1854";
"--ury4=523";
"--llx5=1654";
"--lly5=827";
"--urx5=1854";
"--ury5=1027";
"--filter=60000";
"--ref_level=40000";

```

```

FORS2_chip1_2x2
"--llx=181";
"--lly=7";
"--urx=1861";
"--ury=963";
"--llx1=921";
"--lly1=365";
"--urx1=1121";
"--ury1=565";
"--llx2=181";
"--lly2=7";
"--urx2=381";
"--ury2=207";
"--llx3=181";
"--lly3=763";
"--urx3=381";
"--ury3=963";
"--llx4=1661";
"--lly4=7";
"--urx4=1861";
"--ury4=207";
"--llx5=1661";
"--lly5=763";
"--urx5=1861";
"--ury5=963";
"--filter=62000";
"--ref_level=40000";

```

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 34 of <a href="#">37</a> |

```

FORS2_chip2_1x1
  "--llx=404";
  "--lly=646";
  "--urx=3708";
  "--ury=2054";
  "--llx1=1856";
  "--lly1=1150";
  "--urx1=2256";
  "--ury1=1550";
  "--llx2=404";
  "--lly2=646";
  "--urx2=804";
  "--ury2=1046";
  "--llx3=404";
  "--lly3=1654";
  "--urx3=804";
  "--ury3=2054";
  "--llx4=3308";
  "--lly4=646";
  "--urx4=3708";
  "--ury4=1046";
  "--llx5=3308";
  "--lly5=1654";
  "--urx5=3708";
  "--ury5=2054";
  "--filter=60000";
  "--ref_level=40000";

```

```

FORS2_chip1_1x1
  "--llx=376";
  "--lly=15";
  "--urx=3713";
  "--ury=1924";
  "--llx1=1842";
  "--lly1=730";
  "--urx1=2242";
  "--ury1=1130";
  "--llx2=376";
  "--lly2=15";
  "--urx2=776";
  "--ury2=415";
  "--llx3=376";
  "--lly3=1524";
  "--urx3=776";
  "--ury3=1924";

```

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User<br/>Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 35 of <a href="#">37</a> |

```

"--llx4=3313";
"--lly4=15";
"--urx4=3713";
"--ury4=415";
"--llx5=3313";
"--lly5=1524";
"--urx5=3713";
"--ury5=1924";
"--filter=62000";
"--ref_level=40000";

```

## UT2

- GIRAFFE

```

niter           : 25
tolerance       : 0.1

```

- UVES

```

pix2pix         : TRUE

```

```

RED ARM:
detmon.detmon_opt_lg.exts=-1

```

- X-SHOOTER

```

UVB
"--filter=51000";
"--ref_level=50000";

```

```

VIS
"--filter=53000";
"--ref_level=50000";

```

```

NIR
"--llx=871";
"--lly=709";
"--urx=1065";
"--ury=734";
"--ref_level=10000";

```

## UT3

- ISAAC

|            |  |        |                          |
|------------|--|--------|--------------------------|
| <b>ESO</b> | <b>Detector monitoring recipes User Manual</b> | Doc:   | VLT-MAN-ESO-19500-4846   |
|            |  | Issue: | Issue 1.3.12             |
|            |  | Date:  | Date 2023-03-14          |
|            |  | Page:  | 36 of <a href="#">37</a> |

```
collapse=false
pix2pix=false
```

- VIMOS

```
order          : 2
kappa          : 4.0
llx           : 51
urx           : 2098
pix2pix       : TRUE
tolerance      : 0.1
```

- VISIR Executed with parameter defaults.

## UT4

- HAWK-I

```
exts=-1
```

- NACO

```
ref_level      : 2000
```

- SINFONI

```
detmon.detmon_ir_lg.kappa=9
detmon.detmon_ir_lg.collapse=false
detmon.detmon_ir_lg.pix2pix=true
detmon.detmon_ir_lg.bpmbin=true
```

Currently detmon library do not support VLTI instruments: AMBER, MIDI

VIRCAM is supported running detmon recipe with parameter `-exts=-1`

### 3.2.6 How to keep instrument pipeline and common recipe aligned

Once the linearity data reduction is delegated to a common recipe that is instrument independent each instrument specific corresponding recipe may be removed. This make the corresponding software development independent.

The residual dependency remains at OCA rule levels: each instrument template assigns to the relevant data proper DPR keyword values. The instrument pipeline responsible, in agreement with the corresponding instrument QC scientist, has to keep up to date the OCA rules in order to trigger the common recipe.

|     |   |        |                        |
|-----|---|--------|------------------------|
| ESO | Detector monitoring recipes User Manual | Doc:   | VLT-MAN-ESO-19500-4846 |
|     |   | Issue: | Issue 1.3.12           |
|     |   | Date:  | Date 2023-03-14        |
|     |   | Page:  | 37 of 37               |

The data will be classified with a common tag: RAW\_ON and ROW\_OFF. The data products will be classified accordingly with a common tag (see recipe description).

In order to disentangle products from different instruments the QC scientist may write a small wrapper that changes the products PRO.CATGs to make them instrument dependent (e.g. uses INSTRUME value as prefix). Note that the FITS header of those products may also stay as it is if one classify the products using both PRO.CATG and INSTRUME value. An alternative possible solution, that on the other hand requires to upgrade the current recipe implementation, is to have a recipe input parameter to specify the PRO.CATG, or better, as each recipe have several products, a prefix to be appended to each products to make it instrument dependent.

Due to usual SW development and pipelines and CPL upgrades, there will be often situations, where different instrument accounts use pipeline versions with distinct CPL versions. We expect that the DFS used by DFO is updated more often than every 15 months (usually it is every year and we may expect some delay in the transition to allow QC-Garching to complete operations on previous observing period).

PSD will take care that the shared detmon recipe must be compliant with all operationally used CPL versions (not older than 15 months).

### 3.2.7 How to write for COEFFS\_CUBE product each extension in a separate file

For some instrument, for example OmegaCAM, the COEFFS\_CUBE product may have very large size. This may be a problem for example for the tool that allows to display its content. In case of multi-extension raw data, to limit this problem, the user may set the parameter *coeffs\_cube\_split* to TRUE, in which case are generated as many images as the value of the *order*, each containing in separate extensions results of the corresponding detector. Each image have PRO.CATG set to COEFF\_CUBE\_Pi (i=0...order).

```
detmon.detmon_opt_lg.coeffs_cube_split=TRUE
```

The parameter is FALSE by default. It works for both IR and optical.