edetect_stack

January 26, 2021

Abstract

EPIC source detection on stacked observations, taken at different epochs and / or in mosaic mode.

1 Instruments/Modes

Instrument	Mode
EPIC MOS:	IMAGING
EPIC PN:	IMAGING

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

The meta-task **edetect_stack** performs standardized EPIC source detection on overlapping fields of observations, taken at different epochs or / and in Mosaic Mode. Starting from the filtered event lists, it runs all steps of source detection, produces the necessary input files (images, exposure maps, ...), calls the task **emldetect** for all input images simultaneously, and creates the final stacked source list from its output. Input pointings are considered overlapping if the distance between their centers is **less than 2×12.0 arcmin**.

The handling of **edetect_stack** and its application to a data set are described in Traulsen et al. 2019 and 2020, A&A, arXiv.1807.09178 and arXiv.2007.02932. Users are kindly invited to reference the papers when publishing results based on **edetect_stack**. For a concise description of the XMM-Newton source detection, the 2XMM Catalogue User Guide
http://xmmssc-www.star.le.ac.uk/Catalogue/2XMM/UserGuide_xmmcat.html#SrcDet. can be consulted. Details on the individual tasks are given in their respective documentations (linked below).

Standard input to **edetect_stack** are: one attitude file per observation identifier, one ODF summary file per observation identifier, and all event lists, i.e. up to three event lists per OBS_ID keyword. Attitude and summary files are automatically sorted by OBS_ID. In order to use **edetect_stack** with data taken

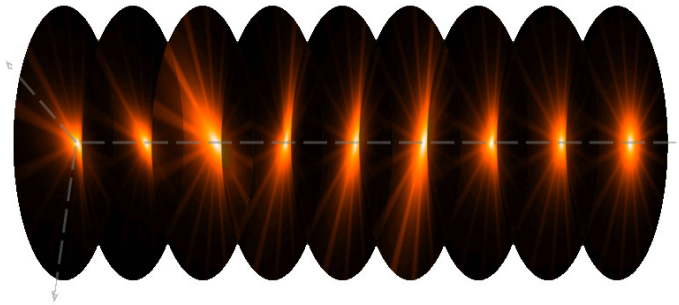

Figure 1: Combining the PSFs of different instruments and observations (offset angles) within the source cut-out radius.

in mosaic mode or on observations which have several event lists for one instrument for any reason, **it is *mandatory* to run the task emosaic_prep on the event lists first in order to introduce a pseudo-exposure ID**. Otherwise, input files may be confused.

edetect_stack comprises twelve stages, which are run subsequently. The task can be stopped and (re-)started at every stage, provided that all input files to the stage are available. Program flow and input parameters are described in more detail in the following subsections.

3.1 General structure

General structure of **edetect_stack**, starting from event lists, attitude information and ODF summary files:

- sort input files and determine the number of different pointings
- set up the common coordinate system: choose reference coordinates and image size from the pointing coordinates and the position angles of the input files (if not explicitly given by the user)
- check whether all imput pointings are overlapping: If the field of view of an input pointing is not overlapping with any other field by at least 3 arcmin in radius, **edetect_stack** exits with an error. If the full fields of view are overlapping, but a pointing is not overlapping with the user-chosen area of interest (reference coordinates & image size), this pointing is ignored, and a warning ist thrown.
- project the input event lists onto the reference coordinate system
- prepare the input files to source detection per pointing (*stages* $2-7$, see below)
- perform stacked source detection on all pointings simultaneously
- create the final source list, which includes information on the detections in the individual pointings and all-EPIC, and a summary source list, which includes one all-EPIC row per valid detection.

Each input image is combined with its respective background image, exposure map, and detection mask. Within a cut-out radius about each tentative detection, the appropriate PSFs are chosen for all valid images, as illustrated in Fig. 1, and source position, counts, and extent are fit simultaneously.

3.2 Input parameters

The input parameters are organized in three groups: general source-detection parameters of the task **edetect_stack**, parameters determining the program flow, and subtask parameters which are passed by **edetect_stack** to the SAS source-detection tasks when calling them.

The most relevant generic parameters of **edetect_stack** are

`eventsets`	–	a blank-separated list of file names of event lists
`attitudesets`	–	a blank-separated list of **atthkgen** attitude files
`summarysets`	–	a blank-separated list of **odfingest** ODF summary files
`ecf`	–	(optionally) the list of energy-conversion factors to calculate valid fluxes
`pimin`, `pimax`	–	(optionally) list of lower and upper limits of the energy bands in eV. Will be used for each pointing-in

If all stages of source detection shall be performed, the first three list parameters are mandatory. The event lists should be sorted by observation IDs, i.e. `eventsets='pnevents(obs1) m1events(obs1) m2events(obs1) m1events(obs2) m2events(obs2) pnevents(obs3) ...'`. The appropriate attitude and summary files are chosen by **edetect_stack** according to their observation identifier, i.e. the OBS_ID keywords of the event lists and attitude files and the "Observation identifier" line in the summary file. The file-naming convention of all files will be described in the next subsection.

Parameters which determine the program flow start with "run" (`runattcalc`, `runevselect`, ...). Alternatively, the parameters `minstage`, `maxstage` can be used to address the program stages by index (indices shown to the left of Figure 2) or by name:

1. `runattcalc`
2. `runevselectimages`
3. `runeexpmap`
4. `runemask`
5. `runeboxdetectlocal`
6. `runesplinemap`
7. `runeboxdetectmap`
8. `runesensmap`
9. `runemosaic`
10. `runeboxdetectstack`
11. `runemldetect`
12. `finalize`

`minstage` and `maxstage` have precedence over the "run..." and "informational" parameters.

Parameters which are directly passed to another task start with a short version of the parameter name (e.g. `att_imagesize`, `emask_withregionset`, `emask_regionset`). Other parameters which change the behaviour of **edetect_stack** have a free name format (e.g. `pimin`, `with_att_imagesize`).

Default values of the parameters are the values used to produce the 3XMM catalogues where applicable. Part of them differ from the default parameter values of the individual tasks.

The energy bands are given as lower limits `pimin` and upper limits `pimax` and are the same for all instruments and pointings. Therefore, users need to specify one `pimin` and `pimax` component per energy band only, not per image (as in **eboxdetect** and **emldetect**.

The Energy conversion factors (ECFs) are used to convert the EPIC count rates to fluxes and are given via the `ecf` parameter in units of 10^{11} counts cm^2 / erg; one ECF per input image, i.e. one value per observation per instrument per energy band. **edetect_stack** uses

- user-supplied values, if the parameter `ecf` is set;
- 3XMM conversion factors, if `ecf` is not set and if standard; energy bands are used
- 0.0 otherwise, which means "NULL" fluxes in the output source list.

A table of all EPIC energy conversion factors is available via the 3XMM-DR5 Catalogue User Guide at http://xmmssc.irap.omp.eu/Catalogue/3XMM-DR5/3XMM-DR5_Catalogue_User_Guide.html#TabNewECFs. *The 3XMM conversion factors are only valid if standard patterns*[1] *are used to produce the images from pipeline event lists.* **edetect_stack** applies the 3XMM patterns by default. More information on ECFs are available via the **emldetect** documentation.

3.3 Tasks called by edetect_stack

edetect_stack consists of twelve subsequent stages during which ten SAS tasks are called to perform the source detection. While running the meta-task, information on the program flow will be shown if `SAS_VERBOSITY` or the argument of the option '-V' is set to 5 or higher.

Figure 2 illustrates the flow of standard processing: tasks and products, and indicates how often each task is called.

Notes on the individual task calls.

1. **attcalc**
   `SAS_ODF` is set for each attitude file according to the input parameter `summarysets` and the observation identifier in the files. The analysis area can be chosen by the user via the parameters `att_nominalra`, `att_nominaldec`, and `att_imagesize` which are forwarded to **attcalc** and activated by `with_att_nominalcoord`=yes and / or `with_att_imagesize`=yes.

   When automatically determining the image area, "notOnChip" warnings may be sent by **esky2det**. They can be ignored. **edetect_stack** informs the user via an extra warning in case an input pointing does not overlap with the chosen analysis area.

2. **evselect**
   By default, EPIC images are filtered with the FLAG and pattern expressions that were used for the 2XMM and 3XMM catalogues (described in the 2XMM Catalogue User Guide). Neither explicit gti filtering nor additional pattern filtering of the input event lists are directly supported by **edetect_stack**. If users want to apply their own filtering expressions, they need to filter the event lists before running **edetect_stack** and/or create their own input images.

   edetect_stack checks whether output images are empty. Instruments / observations for which no image contains any event are ignored in the following.

[1] Cf. http://xmmssc-www.star.le.ac.uk/Catalogue/2XMM/UserGuide_xmmcat.html#ImageCreat

Figure 2: Structure of **edetect_stack**.

3. **eexpmap**
   Exposure maps are created with `withvignetting` (see documentation of eexpmap)=no. Vignetted exposure maps are produced in addition, if `esp_fitmethod` is set to "model (see notes on the call to **esplinemap**).

4. **emask**
   The detector masks are created for the lowest energy band per pointing and instruments and applied to all energy bands. Users are advised to check the automatically created detector masks and to adjust the **emask** thresholds `emask_threshold1`, `emask_threshold1` if, for example, too large regions are masked.

5. **eboxdetect** (local mode)
   Within **edetect_stack**, **eboxdetect** is generally called with exposure maps and detection masks, i.e. the parameters `withexpimage` (see documentation of eboxdetect) and `withdetmask` (see documentation of eboxdetect) are set to "yes" (**eboxdetect** default: "no").

6. **esplinemap**
   edetect_stack supports all fit methods of **esplinemap**, which are explained in the task

description (see documentation of esplinemap):

`esp_fitmethod`=smooth (default of **edetect_stack**): produce background maps via adaptive smoothing of the cheesed image.

`esp_fitmethod`=spline: Use a spline fit to produce the background maps. The number of spline nodes is derived from the image size in pixels, per default, as square root of the longest side divided by two. Users can change the number of nodes by changing the value of the parameter `esp_nsplinenodes` and setting `with_esp_nsplinenodes`=true.

`esp_fitmethod`=model: pass vignetted exposure maps to **esplinemap** via the **esplinemap** parameter `expimageset` (see documentation of esplinemap).
If, additionally, `esp_withexpimageset2`=true: pass unvignetted exposure maps to **esplinemap** via the **esplinemap** parameter `expimageset2` (see documentation of esplinemap).

Users are advised to check the background maps for fit artefacts like very bright regions and to adjust the fit parameters (`esp_nsplinenodes` or `esp_snrmin`, `esp_smoothsigma`) when indicated.

7. **eboxdetect** (map mode)
   The map-mode box source lists per pointing are used as input to the stacked **emldetect** run.

8. **esensmap**
   The creation of sensitivity maps is not mandatory to run the source detection tasks, but recommended for informational purposes. The likelihood threshold `mlmin` (see documentation of esensmap) of **esensmap**, for which the upper limit of source counts is calculated, is set to the minimum detection likelihood of **edetect_stack** (as given via the parameter `mlmin`), **but not below 1.0**.

9. **emosaic**
   The energy-resolved mosaic images are created for informational purposes only. If five energy bands are used – default in the XMM catalogues –, the images of bands 1 and 2 and of bands 4 and 5 are combined in order to serve as input for RGB images.

10. **eboxdetect** (stacked, map mode) with **emosaic** and **srcmatch**

    (a) **emosaic** is called to create mosaicked images, background maps, exposure maps, and detection masks of all pointings per instrument and energy band.

    (b) **eboxdetect** in map mode is run on the *mosaicked* images, i.e. on the same number of input images as in standard source detection, to create a preliminary box source list. Its name derived from the parameter `eboxs_boxlistset` with an inserted "_intermediate". Within **edetect_stack**, **eboxdetect** is generally called with exposure maps and detection masks, i.e. the parameters `withexpimage` and `withdetmask` (see documentation of eboxdetect) are set to "yes" (**eboxdetect** default: "no").

    (c) **srcmatch** combines the intermediate stacked box source list with the map-mode box source lists of the individual pointings (if created in stage 7). The output source list is named `eboxs_boxlistset`.

    The matching radius is determined by **edetect_stack** from the image binning. If users would like to test different matching radii, they may stop **edetect_stack**, adjust the value of the parameter `maxerr` of **srcmatch** and repeat the matching, before restarting **edetect_stack** with `minstage`=11 and the adjusted input list.

    Please note that all the **eboxdetect** source lists that are produced by **edetect_stack** serve as input to other tasks and are not suitable for a quantitative analysis.

11. **emldetect**
    Within **edetect_stack**, the **emldetect** source list is an intermediate product, from which the final stacked source list is created. It is produced with `withrawrows` (see documentation of emldetect)=yes and, by default, with a lower minimum detection likelihood than the final source list.

3.4 Input / output file names

To ease the handling of the file-name parameters, the task **edetect_stack** uses a fixed structure of file names, composing it of a base name like "image" plus extensions per observation, instrument, and energy band (where applicable). The user can provide base names for each file type, which will be expanded by **edetect_stack** following the pre-defined file-name structure. The default file-name structure is defined as follows:

attcalc corrected event lists	–	`events0123456789E000II.fits`
evselect images	–	`iimage0123456789E000II_01234_12345.fits`
eexpmap un-vignetted exposure maps	–	`expunv0123456789E000II_01234_12345.fits`
eexpmap vignetted exposure maps	–	`expmap0123456789E000II_01234_12345.fits`
emask detection masks	–	`detmsk0123456789E000II.fits`
local-mode **eboxdetect** source lists	–	`loclst0123456789E000.fits`
esplinemap background maps	–	`bkgmap0123456789E000II_01234_12345.fits`
esplinemap cheesed images *(diagnostic)*	–	`cheese0123456789E000II_01234_12345.fits`
map-mode **eboxdetect** source lists	–	`boxlst0123456789E000.fits`
emldetect model images *(diagnostic)*	–	`srcmap0123456789E000II_01234_12345.fits`
esensmap sensitivity maps *(diagnostic)*	–	`snsmap0123456789E000II_01234_12345.fits`
emosaic mosaicked images *(diagnostic)*	–	`mosaic_EPIC_01234_12345.fits`, `mosaic_II.fits`

where the first 10 digits denote the observation ID, "E000" an exposure identifier, "II" the instrument, and the following numbers the lower and upper limits of the energy band in eV respectively. Via the parameter `prefix`, an additional identifier can be prepended to *all* file names.

The obvious advantage of the fixed file-name structure is the comprehensive input; its disadvantage the resultant inflexibility regarding input file names. Therefore, lists of file names can be used alternatively to the base names. **edetect_stack** will sort and associate these input files with different pointings and instruments according to their header keywords OBS_ID, EXP_ID, and INSTRUME. **The energy bands, however, cannot be read from the file headers, and standard naming** as <first part of the file name>_<PIMIN>_<PIMAX>, using five digits per PIMIN and PIMAX, **is mandatory.** Example: myinputfile_00500_010000.fits.

The third alternative are ASCII input files, which contain one file name per row. The ASCII file name has to be preceded by an @ sign, as familiar to users from the heatools / FTOOLS tasks. Example: "`eventsets`=@eventlists.txt",

```
...> cat eventlists.txt
obs1_pn_evts.fits
obs1_m1_evts.fits
obs1_m2_evts.fits
obs2_pn_evts.fits
obs2_m2_evts.fits
...>
```

3.5 Stacked output source list

The final output source list comprises all detections of the intermediate **emldetect** source list that have an equivalent detection likelihood `DET_ML` **above** `mlmin` **in total or in at least one pointing**. For each valid detection, it gives one summary line with the combined values of all input observations, one per camera, one per camera and energy band, one all-EPIC summary line per valid pointing and one summary line per instrument per valid pointing. In these summary rows per pointing, the column `ID_INST` includes the observation ID followed by the usual 0 which marks summary rows. They are calculated

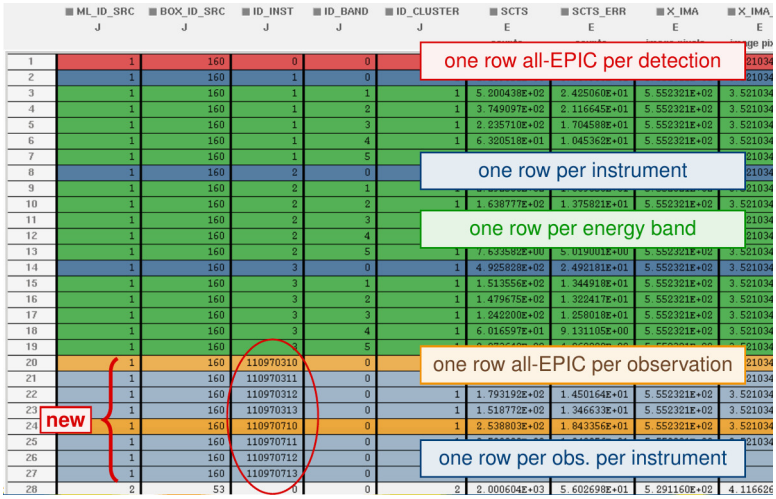

Figure 3: Sketch of the rows per detection in a stacked source list.

Table 1: Default band assignments of hardness ratios HR_i for the EPIC instruments and default energy intervals during pipeline processing (cf. 3XMM-DR5 Catalogue User Guide http://xmmssc.irap.omp.eu/Catalogue/3XMM-DR5/3XMM-DR5_Catalogue_User_Guide.html#TabBands).

i	n	m	Pipeline energy bands [keV]	
1	1	2	0.2–0.5	0.5–1.0
2	2	3	0.5–1.0	1.0–2.0
3	3	4	1.0–2.0	2.0–4.5
4	4	5	2.0–4.5	4.5–12.0

by **edetect_stack** in the same way as the summary rows by **emldetect**, but they are *not* necessarily identical to the results of a separate **emldetect** run on the single pointing. Figure 3 shows a sketch of a stacked source list. A complete list of the table columns is given in Table 2 in Section 8. An additional output source list in the format of the Serendipitous Source Catalogues is named `srclistset`_sum.fits and described in the 3XMM-DR7s catalogue papers by Traulsen et al. (2019, 2020). Since the final stacked source list is created from the intermediate **emldetect** product and detections of low likelihood are rejected, the detection identifiers `ML_ID_SRC` may have gaps, and the maximum `ML_ID_SRC` may be larger than the total number of valid detections.

3.6 Usage hints

At **start-up**, the tasks reads several header keywords of the input files in order to establish the stacking parameters and to sort the input files. The I/O performance of the CFITSIO library, which is used to open the FITS files, strongly depends on the file size. Users are recommended to filter the event lists *before* running **edetect_stack**, if they experience performance issues.

At **runtime**, the performance strongly depends on the number of images to be processed. The number of pointings, which are combined in one call to **edetect_stack** should be limited to the absolutely necessary. Large mosaics should be split into several regions which are processed separately. Correspondingly, users can divide the field of view of repeatedly observed pointings into smaller areas, process them separately and combine the output source lists.

Pseudo exposure IDs set by **emosaic_prep** should be below 110100.

3.7 Examples

The most convenient application:

- create a subdirectory where the task is run and all products are stored
- create ASCII lists of the input files
- call **edetect_stack** and lean back for a while
- check the diagnostic, intermediate, and final data products
- adjust parameters where necessary

Examples for calls to **edetect_stack**:

Process three observations, using default parameters:

```
edetect_stack attitudesets='0110970201/pps/P0110970201OBX000ATTTSR0000.FTZ
                            0110970301/pps/P0110970301OBX000ATTTSR0000.FTZ
                            0110970401/pps/P0110970401OBX000ATTTSR0000.FTZ' \
              summarysets='0130_0110970201_SCX00000SUM.SAS
                           0130_0110970301_SCX00000SUM.SAS
                           0130_0110970401_SCX00000SUM.SAS' \
              eventsets='0110970201/pps/P0110970201M1S002MIEVLI0000.FTZ
                         0110970201/pps/P0110970201M2S003MIEVLI0000.FTZ
                         0110970201/pps/P0110970201PNS001PIEVLI0000.FTZ
                         0110970301/pps/P0110970301M2S003MIEVLI0000.FTZ
                         0110970301/pps/P0110970301PNS001PIEVLI0000.FTZ
                         0110970401/pps/P0110970401M1S002MIEVLI0000.FTZ
                         0110970401/pps/P0110970401M2S003MIEVLI0000.FTZ
                         0110970401/pps/P0110970401PNS001PIEVLI0000.FTZ' \
              -V 5
```

Read file names from ASCII files:

```
edetect_stack attitudesets=@attitudes.txt \
              summarysets=@summaries.txt \
              eventsets=@eventlists.txt \
              -V 5
```

Source detections on previously processed files with standard names, re-setting the minimum detection likelihoods:

```
edetect_stack attitudesets=@attitudes.txt \
              summarysets=@summaries.txt \
              eventsets=@eventlists.txt \
              minstage=8 \
              eboxm_likemin=6 \
              eboxs_likemin=6 \
              eml_mlmin=2 \
              mlmin=10 \
              -V 5
```

Interrupt **edetect_stack** after creating the stacked **eboxdetect** source list in order to insert your favourite objects manually or **srcmatch** another list into the input source list of **emldetect**:

```
edetect_stack attitudesets=@attitudes.txt \
              summarysets=@summaries.txt \
              eventsets=@eventlists.txt \
              maxstage=runeboxdetectstack \
              -V 5
edetect_stack attitudesets=@attitudes.txt \
              summarysets=@summaries.txt \
              eventsets=@eventlists.txt \
              minstage=runemldetect \
              -V 5
```

4 References

5 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
eventsets	no	filename list	event.fits	

Blank-separated name(s) of input event files, from which images and masks are extracted. Alternatively: name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4).
*This parameter is mandatory, if you want to run all stages of the **edetect_stack** source detection.*

attitudesets	no	filename list	attitude.fits	

Blank-separated name(s) of input attitude files, from which the common coordinate system of the stacked observations is derived. One per observation identifier. Alternatively: name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4).
*This parameter is mandatory, if you want to run all stages of the **edetect_stack** source detection.*

summarysets	no	filename list	summary.fits	

Blank-separated name(s) of input odf summary files, as provided by **odfingest**. One per observation identifier. Alternatively: name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4).
*This parameter is mandatory, if you want to run all stages of the **edetect_stack** source detection.*

mlmin	no	float	6.0	[0.0 : 100.0]

Minimum detection likelihood to be reached in total or in at least one pointing to transfer a source from the intermediate to the final output list.

pimin	no	integer	200 500 1000 2000 4500	[0 : 30000]

Lower boundaries of energy bands in source detection. Units: eV

pimax	no	integer	500 1000 2000 4500 12000	[0 : 30000]

Upper boundaries of energy bands in source detection. Units: eV

ecf	no	float	0.0 0.0 0.0 0.0 0.0	[0.0 : 1000.0]

Energy conversion factors, given in units of 10^{11} counts cm^2 / erg, used to derive source fluxes. One value per observation per instrument per energy band. 3XMM values are chosen automatically, if parameter `ecf` is not set and standard energy bands are used.

srclistset	no	filename	srclist.fits	

Name of output **edetect_stack** source list

minstage	no	string	1	[1 : 12] or stage name

Index or name of stage at which **edetect_stack** should be started. If set, it has priority over the "run*" parameters.

maxstage	no	string	12	[1 : 12] or stage name

Index or name of stage at which **edetect_stack** should be stopped. If set, it has priority over the "run*" parameters.

prefix	no	string		

Prefix of all files for which the default name structure is used.

informational	no	string		default | all | none

Write / omit informational output: sensitivity maps, mosaic images (default: yes), cheesed background images and masks, emldetect source images (default: no).

compress	no	boolean	false	true | false

Compress FITSfiles right after creation. Slows down I/O, reduces hard disk usage significantly.

attcalc *(stage 1)*

runattcalc	no	boolean	true	true | false

Run task **attcalc** to establish a common coordinate system. If set to "false", this has to be done manually before starting **edetect_stack**.

att_eventset	no	filename	events	

attcalc: 1. Base name of event lists written by **attcalc**, expanded by **edetect_stack** to <att_eventset> <pointing ID><instrument>.fits, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4). Output files, if `runattcalc` is set to "true" (default), input files to the following tasks otherwise.

with_att_nominalcoord	no	boolean	false	true | false

User input of parameters `att_nominalra` and `att_nominaldec`. If "false": optimum values are derived by **edetect_stack** from the attitude files.

att_nominalra	no	angle	0.0	[0.0 : 360.0]

attcalc: Celestial right ascension coordinate of central reference point, if `with_att_nominalcoord` is set to "true" (default: "false").

att_nominaldec	no	angle	0.0	[-90.0 : +90.0]

attcalc: Celestial declination coordinate of central reference point, if `with_att_nominalcoord` is set to "true" (default: "false").

with_att_imagesize	no	boolean	false	true | false

User input of parameter `att_imagesize`. If "false": optimum value is derived by **edetect_stack** from the attitude files.

att_imagesize	no	angle	0.36	[0.0 :]

attcalc: Image half-size in degrees, if `with_att_imagesize` is set to "true" (default: "false").

evselect *(stage 2)*

runevselectimages	no	boolean	true	true | false

Run task **evselect** to produce images. If set to "false", the files have to be provided as input.

ev_imageset	no	filename	iimage	

evselect: 1. Base name of image files, expanded by **edetect_stack** to <ev_imageset><pointing ID><instrument>_<E_low>_<E_high>.fits, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4). Output files, if `runevselectimages` is set to "true" (default), input files to the following tasks otherwise.

ev_xcolumn	no	string	X	column name

evselect: Name of column with spatial X coordinates for image creation.

ev_ycolumn	no	string	Y	column name

evselect: Name of column with spatial Y coordinates for image creation.

ev_ximagebinsize	no	real	80.0	>0

evselect: Binning factor for the X axis in image creation.

ev_yimagebinsize	no	real	80.0	>0

evselect: Binning factor for the Y axis in image creation.

ev_withxranges	no	boolean	false	true | false

evselect: Use the `ev_ximagemin` and `ev_ximagemax` values for the X coordinate ranges for image creation. If true, the parameter ranges are used; if false, the ranges are determined from the data.

ev_ximagemin	no	real	1.0	

evselect: If set, the lower limit of the X coordinate for image creation.

ev_ximagemax	no	real	1.0	

evselect: If set, the upper limit of the X coordinate for image creation.

ev_withyranges	no	boolean	false	true | false

evselect: Use the `ev_yimagemin` and `ev_yimagemax` values for the Y coordinate ranges for image creation. If true, the parameter ranges are used; if false, the ranges are determined from the data.

ev_yimagemin	no	real	1.0	

evselect: If set, the lower limit of the Y coordinate for image creation.

ev_yimagemax	no	real	1.0	

evselect: If set, the upper limit of the Y coordinate for image creation.

ev_withimagedatatype	no	boolean	true	true | false

evselect: Boolean to choose whether to use the value of the `ev_imagedatatype` parameter to set the data type of the output image; if the value is false then **evselect** tries itself to determine the best value for the image data type. *Please note: "stand-alone" **evselect** sets this parameter to "false" by default, **edetect_stack** to "true".*

ev_imagedatatype	no	string	Int32	Int8 | Int16 | Int32 | Real32 | Real64

Data type to use for the output image. If not set, **evselect** decides for itself what data type to use. *Please note: "stand-alone" **evselect** sets this parameter to "Real64" by default, **edetect_stack** to "Int32".*

eexpmap *(stage 3)*

runeexpmap	no	boolean	true	true | false

Run task **eexpmap** to produce exposure maps. If set to "false", the files have to be provided as input.

eexp_expimageset	no	filename	expmap	

eexpmap: 1. Base name of *unvignetted* exposure image files, expanded by **edetect_stack** to <eexp_expimageset> <pointing ID><instrument>_<E_low>_<E_high>.fits, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4). Output files, if runeexpmap is set to "true" (default), input files to the following tasks otherwise.

eexp_attrebin	no	float	2.0	[0.0 : 60.0]

eexpmap: Positional accuracy of attitude rebinning in arcseconds. Changes in the attitude less than eexp_attrebin are ignored when rebinning the attitude data. *Please note: "stand-alone" **eexpmap** sets this parameter to 4.0 by default, **edetect_stack** to 2.0.*

emask *(stage 4)*

runemask	no	boolean	true	true | false

Run task **emask** to produce detection masks. If set to "false", the files have to be provided as input.

emask_detmaskset	no	filename	detmsk	

emask: 1. Base name of detection mask files, expanded by **edetect_stack** to <emask_detmaskset> <pointing ID><instrument>.fits, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4). Output files, if runemask is set to "true" (default), input files to the following tasks otherwise.

emask_threshold1	no	float	0.5	[0.0 : 1.0]

emask: Threshold parameter 1: fraction of maximum exposure. *Please note: "stand-alone" **eexpmap** sets this parameter to 0.3 by default, **edetect_stack** to 0.5.*

emask_threshold2	no	float	1.0	[0.0 : 10.0]

emask: Threshold parameter 2: threshold for gradient of exposure. *Please note: "stand-alone" **emask** sets this parameter to 0.5 by default, **edetect_stack** to 1.0.*

emask_withregionset	no	boolean	false	true | false

emask: Read a FITS region file and exclude circular/box regions from the mask.

emask_regionset	no	filename	emask.fits	

emask: Name of the FITS region file providing the regions to be excluded, if emask_withregionset is set to true.

eboxdetect (local mode) *(stage 5)*

runeboxdetectlocal	no	boolean	true	true | false

Run task **eboxdetect** to produce local-mode input source lists per observation. If set to "false", the files have to be provided as input.

eboxl_boxlistset	no	filename	eboxlocal.fits	loclst

eboxdetect: 1. Base name of local-mode box list files, expanded by **edetect_stack** to <ev_imageset><pointing ID>.fits, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4). Output files, if runeboxdetectlocal is set to "true" (default), input files to the following tasks otherwise.

eboxl_likemin	no	float	5.0	[1.0 : 50.0]

eboxdetect: Minimum detection likelihood for including a source in the output list. *Please note: "stand-alone" **eboxdetect** sets this parameter to 10.0 by default, **edetect_stack** to 5.0.*

eboxl_boxsize	no	integer	5	3 | 5

eboxdetect: Size of detection box (3x3 or 5x5 pixels).

eboxl_nruns	no	integer	1	[1 : 4]

eboxdetect: Number of detection runs (detection box size is doubled after each run). *Please note: "stand-alone" **eboxdetect** sets this parameter to 3 by default, **edetect_stack** to 1.*

esplinemap *(stage 6)*

runesplinemap	no	boolean	true	true | false

Run task **esplinemap** to produce spline background maps. If set to "false", the files have to be provided as input.

esp_bkgimageset	no	filename	bkgmap	

esplinemap: 1. Base name of background map files, expanded by **edetect_stack** to <esp_bkgimageset><pointing ID><instrument>_<E_low>_<E_high>.fits, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4). Output files, if runesplinemap is set to "true" (default), input files to the following tasks otherwise.

esp_scut	no	float	0.002	[0.0 : 10.0]

esplinemap: Source cut-out flux level in [counts/arcsec^2]. *Please note: "stand-alone" **esplinemap** sets this parameter to 0.01 by default, **edetect_stack** to 0.002.*

esp_mlmin	no	float	1.0	[0.0 : 1000.0]

esplinemap: Minimum single band detection likelihood for sources to be cut out.

esp_fitmethod	no	string	spline	spline | model

esplinemap: Background fitting method: spline fit or 2-component background model.

with_esp_nsplinenodes	no	boolean	false	true | false

User input of parameter esp_nsplinenodes. If "false": A standard value of one node per 50 pixels is derived by **edetect_stack** from the image size.

esp_nsplinenodes	no	integer	13	[10 : 40]

esplinemap: Number of nodes in spline fit. *Please note: "stand-alone" **esplinemap** sets this parameter to 13.0 by default. **edetect_stack** uses a default value that depends on image size in pixels.*

esp_excesssigma	no	float	4.0	[1.0 : 6.0]

esplinemap: Threshold for sigma excesses with respect to background spline fit.

esp_nfitrun	no	integer	4	[1 : 5]

esplinemap: Number of iterations for removal of excesses. esp_nfitrun=1 means no removal. *Please note: "stand-alone" **esplinemap** sets this parameter to 3 by default, **edetect_stack** to 4.*

esp_snrmin	no	float	30.0	[1.0 : 1000.0]

esplinemap, fitmethod smooth: Desired minimum signal to noise ratio during adaptive smoothing.

esp_smoothsigma	no	float	15	[0.0 : 100.0]

esplinemap, fitmethod smooth: Minimum width of Gaussian smoothing kernel in pixel.

esp_withexpimageset2	no	boolean	false	true | false

esplinemap: Use both vignetted and unvignetted exposure maps for fitmethod=model.

esp_expimagesetvig	no	filename	expvig	

esplinemap: 1. Base name of *vignetted* exposure maps for fitmethod=model, expanded by **edetect_stack** to `<esp_expimagesetvig><pointing ID><instrument>_<E_low>_<E_high>.fits`, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4).

esp_withcheeseimage	no	boolean	true	true | false

esplinemap: Optional output of "cheesed" photon images where sources have been masked out. *Please note: "stand-alone"* ***esplinemap*** *sets this parameter to "false" by default,* ***edetect_stack*** *to "true".*

esp_cheeseimageset	no	filename	cheese	

esplinemap: 1. Base name of diagnostic output cheesed image files, expanded by **edetect_stack** to `<esp_cheeseimageset><pointing ID><instrument>_<E_low>_<E_high>.fits`, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4).

esp_withcheesemask	no	boolean	false	true | false

esplinemap: Optional output of a cheese-mask images. Values 0 stand for masked areas, 1 for valid image areas.

esp_cheesemaskset	no	filename	cheesemask	

esplinemap: 1. Base name of diagnostic output cheese mask image files, expanded by **edetect_stack** to `<esp_cheesemaskset><pointing ID><instrument>_<E_low>_<E_high>.fits`, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4).

eboxdetect (map mode) *(stage 7)*

runeboxdetectmap	no	boolean	true	true | false

Run task **eboxdetect** to produce map-mode input source lists per observation. If set to "false", the files have to be provided as input.

eboxm_boxlistset	no	filename	eboxmocal.fits	boxlst

eboxdetect: 1. Base name of map-mode box list files, expanded by **edetect_stack** to `<ev_imageset><pointing ID>.fits`, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4). Output files, if `runeboxdetectmap` is set to "true" (default), input files to the following tasks otherwise.

eboxm_likemin	no	float	5.0	[1.0 : 50.0]

eboxdetect: Minimum detection likelihood for including a source in the output list. *Please note: "stand-alone"* ***eboxdetect*** *sets this parameter to 10.0 by default,* ***edetect_stack*** *to 5.0.*

eboxm_boxsize	no	integer	5	3 | 5

eboxdetect: Size of detection box (3x3 or 5x5 pixels).

eboxm_nruns	no	integer	3	[1 : 4]

eboxdetect: Number of detection runs (detection box size is doubled after each run).

eboxm_hrdef	no	integer	1 2 2 3 3 4	[0 : 10]

eboxdetect: Array of six integer numbers specifying the upper and lower energy band for each of (up to) three hardness ratios.

esensmap *(stage 8)*

runesensmap	no	boolean	true	true | false

Run task **esensmap** to produce sensitvity maps.

esen_sensimageset	no	filename	snsmap	

esensmap: 1. Base name of sensitivity image files, expanded by **edetect_stack** to <esen_sensimageset><pointing ID><instrument>_<E_low>_<E_high>.fits, or 2. a list of file names, or 3. name of an ASCII file preceeded by '@', containing one file name per line (cf. 3.4).

emosaic *(stage 9)*

runemosaic	no	boolean	true	true | false

Run task **emosaic** to produce a mosaic image for illustrative purposes.

emos_mosaicedset	no	filename	mosaic	

emosaic: Base name of illustrative mosaic images, expanded by **edetect_stack** to <emos_mosaicedset>_EPIC_<E_low>_<E_high>.fits and <emos_mosaicedset>_<instrument>.fits.

eboxdetect (map mode, stacked) *(stage 10)*

runeboxdetectstack	no	boolean	true	true | false

Run task **eboxdetect** in map mode to produce stacked input source lists. If set to "false", the file has to be provided as input.

eboxs_boxlistset	no	filename	eboxlist.fits	

eboxdetect: Name of stacked box source list. Output file, if runeboxdetectstack is set to "true" (default), input file otherwise.

eboxs_likemin	no	float	5.0	[1.0 : 50.0]

eboxdetect: Minimum detection likelihood for including a source in the output list. *Please note: "stand-alone" **eboxdetect** sets this parameter to 10.0 by default, **edetect_stack** to 5.0.*

eboxs_boxsize	no	integer	5	3 | 5

eboxdetect: Size of detection box (3x3 or 5x5 pixels).

eboxs_nruns	no	integer	3	[1 : 4]

eboxdetect: Number of detection runs (detection box size is doubled after each run).

eboxs_hrdef	no	integer	1 2 2 3 3 4	[0 : 10]

eboxdetect: Array of six integer numbers specifying the upper and lower energy band for each of (up to) three hardness ratios.

emldetect *(stage 11)*

runemldetect	no	boolean	true	true | false

Run task **emldetect** to produce the stacked output source list.

eml_mllistset	no	filename	emllist.fits	

emldetect: Name of intermediate stacked source list.

eml_psfmodel	no	string	ellbeta	ellbeta | medium | slew

emldetect: Model PSF: fully 2d parameterized analytical or medium-accuracy EPIC PSFs.
Users are strongly encouraged to use the new ellbeta PSF for which this task was designed.

eml_mlmin	no	float	0.001	[0.0 : 100.0]

emldetect: Minimum detection likelihood to include a source in the intermediate source list. *Please note: "stand-alone"* ***emldetect*** *sets this parameter to 10.0 by default,* ***edetect_stack*** *to 0.001 in order to create the intermediate source list.*

eml_dmlextmin	no	float	4.0	[0.0 : 100.0]

emldetect: Minimum likelihood to regard a source as extended. *Please note: "stand-alone"* ***emldetect*** *sets this parameter to 10.0 by default,* ***edetect_stack*** *to 4.0.*

eml_fitextent	no	boolean	true	true | false

emldetect: Fit source extent. *Please note: "stand-alone"* ***emldetect*** *sets this parameter to "false" by default,* ***edetect_stack*** *to "true".*

eml_minextent	no	float	1.5	[0.0 : 300.0]

emldetect: Minimum allowed value for the extent parameter of an extent model in image pixels.

eml_maxextent	no	float	20.0	[0.1 : 300.0]

emldetect: Maximum allowed value for the extent parameter of an extent model in image pixels.

eml_extentmodel	no	string	beta	gaussian | beta

emldetect: Model function for source extent, if `eml_fitextent` is set to "true". *Please note: "stand-alone"* ***emldetect*** *sets this parameter to "gaussian" by default,* ***edetect_stack*** *to "beta".*

eml_nmaxfit	no	integer	1	[1 : 10]

emldetect: Maximum number of neighbouring sources to be fit simultaneously (`eml_nmaxfit`$*$`eml_nmulsou` ≤ 10).

eml_nmulsou	no	integer	2	[1 : 3]

emldetect: Allow fit to split up one input source in maximum nmulsou sources (`eml_nmaxfit`$*$`eml_nmulsou` ≤ 10). *Please note: "stand-alone"* ***emldetect*** *sets this parameter to 1 by default,* ***edetect_stack*** *to 2.*

eml_fitposition	no	boolean	true	true | false

emldetect: Fit source positions.

eml_determineerrors	no	boolean	true	true | false

emldetect: Determine statistical errors.

eml_ecut	no	float	15.0	[0.4 : 100.0]

emldetect: Source cut-out radius for PSF-fitting. Values lower than 1.0: cut-out radius expressed as fraction of the normalized PSF. Values larger than 1.0 are interpreted as a fixed event cut-out radius given in units of image pixels. *Please note: "stand-alone"* ***emldetect*** *sets this parameter to 0.68 by default,* ***edetect_stack*** *to 15.0.*

eml_scut	no	float	0.9	[0.4 : 100.0]

emldetect: Source selection radius for multi-source fitting. Values lower than 1.0: selection radius ex-

pressed as fraction of the normalized PSF. Values larger than 1.0 are interpreted as a fixed event cut-out radius given in units of image pixels.

eml_withsourcemap	no	boolean	false	true | false

emldetect: Produce simulated source maps.

eml_sourceimagesets	no	filename	srcmap	

emldetect: Base name of output source map files, if `eml_withsourcemap` is set to "true", expanded by **edetect_stack** to <`eml_sourceimageset`><`pointing ID`><`instrument`>_<`E_low`>_<`E_high`>`.fits`.

eml_hrpndef	no	integer	1 2 2 3 3 4 4 5	[0 : 10]

emldetect: Array of up to eight indices (integer) specifying the upper and lower energy band for each of the hardness ratios for PN; i.e. two numbers per energy band.

eml_hrm1def	no	integer	1 2 2 3 3 4 4 5	[0 : 10]

emldetect: Array of up to eight indices (integer) specifying the upper and lower energy band for each of the hardness ratios for MOS1; i.e. two numbers per energy band.

eml_hrm2def	no	integer	1 2 2 3 3 4 4 5	[0 : 10]

emldetect: Array of up to eight indices (integer) specifying the upper and lower energy band for each of the hardness ratios for MOS2; i.e. two numbers per energy band.

eml_withthreshold	no	boolean	true	true | false

emldetect: Allow splitting up into multi-PSF fitting only for sources above threshold. *Please note: "stand-alone"* ***emldetect*** *sets this parameter to "false" by default,* ***edetect_stack*** *to "true".*

eml_threshold	no	float	10.0	>0.0

emldetect: Value of threshold for multi-PSF fitting, if `eml_withthreshold` is set to true. *Please note: "stand-alone"* ***emldetect*** *sets this parameter to 20.0 by default,* ***edetect_stack*** *to 10.0.*

eml_threshcolumn	no	string	LIKE	LIKE | SCTS | RATE

emldetect: Column in input list on which `eml_threshold` will be applied, if `eml_withthreshold` is set to true.

eml_withhotpixelfilter	no	boolean	false	true | false

emldetect: If true, the likelihood contribution of the brightest pixel will be ignored (i.e., detections relying on a single pixel will be disregarded).

eml_withtwostage	no	boolean	true	true | false

emldetect: Use two-stage process for multi PSF (`eml_nmulsou` >1) fitting. *Please note: "stand-alone"* ***emldetect*** *sets this parameter to "false" by default,* ***edetect_stack*** *to "true".*

eml_withxidband	no	boolean	false	true | false

emldetect: Write XID band output for the X-ray follow-up and identification programme.

eml_xidfixed	no	boolean	false	true | false

emldetect: Run **emldetect** on XID-band image with positions and source extent fixed to input values.

eml_xidecf	no	float	0.0	[0.0 : 1000.0]

emldetect: XID-band energy conversion factors, given in units of 10^{11} counts cm^2 / erg.

eml_xidpndef	no	integer	2 3 4	[0 : 10]

emldetect: Index of the energy band(s) from which the images / start values for the XID band are taken for PN.

eml_xidm1def	no	integer	2 3 4	[0 : 10]

emldetect: Index of the energy band(s) from which the images / start values for the XID band are taken for MOS1.

eml_xidm2def	no	integer	2 3 4	[0 : 10]

emldetect: Index of the energy band(s) from which the images / start values for the XID band are taken for MOS2.

Final stacked source list *(stage 12)*

finalize	no	boolean	true	true | false

Calculate the final stacked source list from the intermediate emldetect source list.

6 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

NotOverlapping *(error)*
One or more pointings do(es) not overlap with any other input pointing (maximum distance between image centers: 2×12.0 arcmin).

FileMismatch *(error)*
Number of input event lists does not match number of input attitude files.

ParameterError *(error)*
Unknown value of a parameter.

ParameterMismatch *(error)*
Conflict between input parameters (explained in the error message).

ParameterCountMismatch *(error)*
Number of upper energy-band boundaries does not match number of lower energy-band boundaries.

CIFMissing *(error)*
Environment variable `SAS_CCF` is not set.

FileNotFound *(error)*
Could not open input file. File not found or unreadable.

FileNotCopied *(error)*
Could not copy the file to the desired location.

KeywordMissing *(error)*
Mandatory keyword not found in the header of an input file.

EmptySourceList *(error)*
Input source list to emldetect is empty.

TaskError *(error)*
Task called by edetect_stack ended in error.

KeywordMismatch *(warning)*
Header keywords of input files are not consistent. Products of the following tasks may be incorrect.
corrective action: Check input/output files.

KeywordNotChanged *(warning)*
Header keyword could not be changed.
corrective action: Check order and headers of input files.

ImageNotFound *(warning)*
An image file that should have been created or input is missing.
corrective action: Check files and image size / reference coordinates.

InstrumentIgnored *(warning)*
All images for an instrument are empty in an input pointing.
corrective action: Respective file names are removed from the list of input images

PointingIgnored *(warning)*
A pointing is not overlapping with the analysis region.
corrective action: No further processing of the corresponding files.

EmptySourceList *(warning)*
Intermediate eboxdetect source contains no detections.
corrective action: Skipping it.

SinglePointing *(warning)*
No need to produce mosaic images.
corrective action: Skipping stage "runeboxdetectstack".

7 Input Files

Default usage:

1. PPS product by task **atthkgen**: ATTHK FITS file containing attitude information (per observation)
2. PPS product e.g. by meta-tasks **epchain** / **emchain** or **epproc** / **emproc**: event lists (per observation and instrument)
3. PPS product by task **odfingest**: ODF summary file (per observation)

Alternative usage, when running certain stages of **edetect_stack**:

1. PPS products or intermediate products of former calls to **edetect_stack**: input files for subsequent task stages (per pointing, instrument, and energy band). E.g. images, exposure maps.

8 Output Files

1. stacked EPIC **edetect_stack** source list
2. intermediate output of individual tasks called by **edetect_stack** (cf. Subsection 3.3)
3. optionally: optional output of individual tasks called by **edetect_stack** (cf. Subsection 3.3)

Table 2: Columns of the stacked source table

	Name	Description	Unit	Format	Stacked
1	ML_ID_SRC	unique detection number		int.	
2	BOX_ID_SRC	corresponding **eboxdetect** input source number		int.	
3	ID_INST	instrument ID. 0: all-EPIC, 1: PN, 2: MOS1, 3: MOS2, preceded by the observation identifier in pointing-summary rows.		int.	
4	ID_BAND	energy band number (0: total)		int.	
5	ID_CLUSTER	cluster id of detections which have been fit simultaneously		int.	
6	SCTS	background-corrected counts under the entire PSF of the detection	cts	real	total
7	SCTS_ERR	source counts error	cts	real	total
8	X_IMA	position of the detection in X image direction	image px	real	combined fit
9	X_IMA_ERR	error of X_IMA, corresponding to a 1σ-error in chi-squared statistics	image px	real	combined fit
10	Y_IMA	position of the detection in Y image direction	image px	real	combined fit
11	Y_IMA_ERR	error of Y_IMA, corresponding to a 1σ-error in chi-squared statistics	image px	real	combined fit
12	EXT	extent radius of the detection in image pixels, i.e. gaussian sigma or beta model core radius	image px	real	combined fit
13	EXT_ERR	statistical error of the extent	image px	real	combined fit
14	DET_ML	detection likelihood, normalized to two degrees of freedom		real	equiv. likelihood
15	EXT_ML	likelihood of the detection to be extended		real	combined fit
16	BG_MAP	background at the position of the detection	cts/px	real	total
17	EXP_MAP	vignetting corrected exposure of the detection: PSF-weighted mean of the subimages (radius CUTRAD) about the detection	s	real	total
18	FLUX	flux under the entire PSF of the detection	erg/cm^2/s	real	weighted sum
19	FLUX_ERR	statistical error of the flux	erg/cm^2/s	real	weighted error
20	RATE	count rate under the entire PSF of the detection, corrected for background, vignetting, detector efficiency and gaps	cts/s	real	total
21	RATE_ERR	statistical error of the count rate	cts/s	real	total
22	RA	right ascension of the detection	degrees	dble.	combined fit
23	DEC	declination of the detectino	degrees	dble.	combined fit
24	RADEC_ERR	combined R.A.-Dec. statistical error ($\sqrt{\sigma_\alpha^2 + \sigma_\delta^2} = \sqrt{2}\,\sigma_{1\mathrm{d}}$)	arcsec	real	combined fit
25	LII	galactic longitude of the detection	degrees	dble.	combined fit
26	BII	galactic latitude of the detection	degrees	dble.	combined fit

	Name	Description	Unit	Format	Stacked
27	RAWX	raw-X coordinate of the detection	px	int.	per image only
28	RAWY	raw-Y coordinate of the detection	px	int.	per image only
29	OFFAX	off-axis angle	arcmin	real	per image only
30	CCDNR	chip number		int.	per image only
31, 33, 35, 37	HRi ($i \in [1,4]$)	hardness ratio of count rates: HR(band m , band n) = (rate(n)−rate(m)) / (rate(n)+rate(m)). No energy conversion factors are applied.		real	derived from the total rates
32, 34, 36, 38	HRi_ERR ($i \in [1,4]$)	statistical error of HRi		real	of HRi
39	CUTRAD	source cut out radius	image px	real	combined fit
40	MASKFRAC	PSF weighted on-chip fraction		real	max. of minima
41	EEF	encircled energy fraction		real	combined fit
42	VIGNETTING	Vignetting as function of off-axis angle and energy; only valid in the intermediate **emldetect** source list.		real	NULL
43	ONTIME	Integration time of the CCD, not vignetting corrected. NULL, if CCD no. is not defined (i.e. source center on bad pixels, gaps, damaged CCDs)	s	real	total
44	PILEUP	pile-up level of the detection in the active instrument configuration		real	maximum
45	DIST_NN	distance to nearest neighbour	arcsec	real	combined fit
46	FLAG	quality flag placeholder (to be set by **dpssflag**)		char.	–

9 Algorithm

```
sub edetect_stack {

    # prepare
    read parameters and files
    choose energy conversion factors
    define pointing ids, using header keywords
    check whether pointings are overlapping
    derive reference coordinate system and actual image size

    # SOURCE-DETECTION FILES PER OBSERVATION
    loop over observations

        # set reference coordinate system,  if "runattcalc" is set
        set summary file
        copy event files to working directory
        call attcalc per instrument

        # produce images,  if "runevselectimages" is set
        define filtering expression
        call evselect per instrument per energy band

        # produce exposure maps,  if "runeexpmap" is set
```

```
        call eexpmap without vignetting correction per instrument per energy band
        if esplinemap fitmethod is "model"
            call eexpmap with vignetting correction per instrument per energy band
        end if

        # produce detection masks,  if "runemask" is set
        choose energy band of exposure image
        call emask per instrument

        # produce local-mode eboxdetect source list,  if "runeboxdetectlocal" is set
        determine minimum imagebuffersize from header keywords
        call eboxdetect in local mode

        # produce background maps,  if "runesplinemap" is set
        choose fitmethod
        call esplinemap per instrument per energy band

        # produce map-mode eboxdetect source list,  if "runeboxdetectstack" is set
        determine minimum imagebuffersize from header keywords if not yet known
        call eboxdetect in map mode

        # produce sensitivity maps,  if "runesensmap" is set
        call esensmap per instrument per energy band

        # produce informational mosaic images,  if "runemosaic" is set
        call emosaic in different constellations

    end loop over pointing ids

    # STACKED SOURCE DETECTION

    # produce map-mode eboxdetect source list,  if "runeboxdetectstack" is set
    create mosaic images per instrument and energy band
    call eboxdetect in map mode on mosaicked input files
    srcmatch the results with the map-mode eboxdetect lists per pointing

    # produce intermediate emldetect source list,  if "runemldetect" is set
    determine minimum imagebuffersize from header keywords
    call emldetect

    # create final source list,  if "finalize" is set
    sub stack_sourcelist {
        calculate summary lines per pointing and in total
        keep only detections whose detection likelihood is above mlmin
            (in total or in at least one pointing)
        write final fits table
    }

}
```

10 Comments

11 Future developments

References