emevents

February 1, 2016

Abstract

Define the final list of events, correct position and time for one node of one EPIC-MOS CCD over one exposure.

1 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING, TIMING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

emevents flags unvalid events (following **evatt**), subtracts bad pixels, merges events split over two nodes, separates piled up diagonal events into two, adds the time column and monitors the GATTI performance. It also modifies the `RAWX` and `RAWY` coordinates to center the events and computes camera coordinates. **emevents** modifies the number of events. Therefore after it the event file is not consistent with the `NVALID` field of the frame file any more. It may be applied as is to slew data.

As a preliminary step, **emevents** rejects events with invalid `FRAME`, `RAWX`, `RAWY` and `PATTERN`. Those are the result of telemetry errors. They are removed from the list altogether (not only flagged).

emevents calls (in order) the following subroutines, all of which can be individually switched off (this is provided for calibration and debugging activities, the result will not in general be scientifically useful when not used with the default settings):

- EV_REC. In EDU Threshold mode (associated to telemetry REDUCED IMAGING mode), all pixels above the EMDH lower threshold and below the EMDH upper threshold are transmitted individually (no event recognition is performed on board). EV_REC performs the same pattern analysis (Fig. 1) as on-board, with the difference that it is performed with the

EDU threshold set at the EMDH lower threshold (everything below that was lost) and the energies E_3 and E_4 cannot be recovered.
EV_REC generates the `ENERGYE1` and `ENERGYE2` columns (per event) from the original `ENGYE1E2` column (per pixel). It flags the events outside the EMDH thresholds with `OUTSIDE_THRESHOLDS`.
In Imaging mode this pattern analysis is performed after CUT_BAD if `analysepatterns` is set to True.
In Timing mode this pattern analysis is used to reject events classified as singles or bipixels by the 1-D pattern analysis on board (Fig. 2), but which are actually part of a large track from a cosmic-ray in the framestore area. A 2-D pattern analysis, similar to that in Imaging mode (Fig. 1), is run locally around all events with `PATTERN` 0 or 1 and the event is accepted only if it would have passed the Imaging pattern analysis, with any `PATTERN` less than 30. Otherwise it is flagged as `COSMIC_RAY`. No such cosmic-ray rejection is performed if `analysepatterns` is set to False. This may not be applied in Compressed Timing mode because most events do not have a valid `RAWX` and `PATTERN` in that mode.
If Timing mode was configured on-board with `EDUMODE`=3 and Imaging patterns (non-standard), then no additional pattern analysis is carried out within **emevents**. Instead the `PATTERN` values are converted to Timing patterns (Fig. 2) (between 0 and 3), and all events with vertical extension are flagged as `CLOSE_TO_BRIGHTPIX`. Those are true events, but their energy is wrong because it was added to that of another event.

- CUT_BAD. On-board all events centered on a bright pixel (declared in the on-board library) are rejected, but weak bad pixels (with low charge contents) can also be included in X-ray events next to them. Those events must be corrected for that. To that end all areas around bright pixels (declared or not in the on-board library), and next to borders, may be reanalysed, putting the bright pixels and the pixels outside the declared window to 0. In the process events corresponding to bright pixels not declared on board are flagged as `ON_BADPIX`. If a bright pixel is the main pixel of a split event, it may appear twice in output: the original event (flagged as `ON_BADPIX`) plus what remains after removing the bright pixel. Bad pixels are read from the bad pixels extension added by **badpix**. Dead pixels are treated separately. Events on a pixel declared dead are flagged as `ON_BADPIX` whatever their `PATTERN`. In TIMING mode there are no bad pixels, only bad columns. The events are not reanalysed, but simply flagged with `ON_BADPIX`. The same happens in Imaging mode when `analysepatterns` is set to False. This may not be applied in Compressed Timing mode because most events do not have a valid `RAWX` in that mode.

- BINODAL. If the CCD was read through both nodes (mode PRI PART W6 IMAGING only), X-rays interacting close to the median column of the CCD have their charge cloud split over both nodes. BINODAL recognizes as only one event those which have been detected on both sides, using the same patterns as on-board (Fig. 1). This means also rejecting events which become larger than the accepted patterns. Only events with their maximum on this side of the border are kept in the list (*i.e.* the event list remains formally associated to one node). If the frame for the other node is flagged as bad, then its events should not be merged. Instead events immediately next to the other half of the CCD should be flagged as `CLOSE_TO_CCD_BORDER`.

- DIAGO. Diagonal patterns (26 – 29 on Fig. 1) are mostly due to two close-by single events. DIAGO forms two events out of one in that case. Both events are flagged with `DIAGONAL` to distinguish them from true single events. The second event is created only if its charge (*i.e.* E_2) is above the EMDH lower threshold. This operation does not respect the order of increasing `RAWY` within a frame. This routine is not applicable in TIMING mode.
When either E_1 or E_2 is within one HWHM of the energy of Si K (1740 eV), the event can be due to partial absorption associated with Si fluorescence, caught in a neighbouring pixel. This appears as a line at that energy in the spectrum of diagonals. If `keepsifluor` is set to True, these events are not split in two but their `PATTERN` is changed from diagonal to bipixel (so that it receives the appropriate calibration), and it is flagged with `DIAGONAL` (to

remember it was). As a side effect this creates a hole in the spectrum of diagonals at the energy of 1740 eV.

- CUT_GTI. All events whose frames were flagged for rejection by **emframes** (flag above 64) are flagged for rejection with `IN_BAD_FRAME`. Events whose frames were flagged below 64 by **emframes** are flagged with `IN_SPOILED_FRAME`. "cosmic-ray" events (patterns 30 and 31 in Fig. 1, read from `CRPATi` keywords) are flagged with `COSMIC_RAY`. CUT_GTI propagates the `FRMTIME` and `CRPATi` keywords.

- PUT_XY centers the events (telemetered positions are those of the upper-right corner of the 5x5 pattern area), subtracts the underscan (not relevant to the user) and adds 1 (for coherence with the FITS standard on images) so that coordinates run from 1 to 600. It also subtracts the underscans and adds 1 to the `WINDOWX0` and `WINDOWY0` keywords for coherence. Finally, it subtracts the under + overscans (in full frame mode) or 2 (in window mode) to the `WINDOWDX` and `WINDOWDY` keywords.
  It then computes the camera coordinates `DETX`, `DETY`, in the CAMCOORD2 coordinate system as described in **cal**, in units of 0.05 arcseconds. This includes by default randomisation (within a CCD pixel) to avoid Moiré effects. Randomisation may be switched off using `randomizeposition`=N. In TIMING mode the missing coordinates are replaced by the `SRC_RAWX`, `SRC_RAWY` keywords (or the center of the CCD if those keywords do not exist) for computing the camera coordinates, and double events (always centered on the left pixel) are moved half a pixel to the right.
  PUT_XY flags the (false) events outside the window with `OUT_OF_CCD_WINDOW`.

- PUT_TI (IMAGING mode) copies for each event of a given frame the same time computed as the middle between the start and end times of the current frame. Optionally time randomisation (within a frame) may be performed using `randomizeposition`=Y.
  PUT_TI (or PUT_TT) creates and fills the `EXPOSURE` extension, with precise time and duration of each frame and the effective fraction of observing time, corrected for dead-time using the `CRRATIO` column of the frame file. The (small) frame transfer time (during which the sky is actually observed but smeared over a column) is subtracted from the frame duration `TIMEDEL`. Following OGIP/93-003, the time is that of the center of each frame (*i.e.* the same as that of the events).

- PUT_TT (TIMING mode) computes the time for each event as a reference time (the start time of the current cycle) plus (y_i + 0.5) times the clocking time. Precise corrections on the basis of the source position on the CCD are not done at this level. Times are computed as if the source was at the center of the CCD.

- REJECT_ROWS. Some instrumental effects result in several events following one another in the same row in one frame (particularly at small `RAWY`), while this is unlikely for normal X-rays. A simple way to reject this noise is to reject all events (of that frame) found in rows with many events in a given frame. This is done with flag `ON_BADROW`. The threshold may be set via `maxeventsperrow`.
  To avoid rejecting true events for bright extended sources or strongly piled-up sources, the average number of counts in that row plus twice its square root (2 σ security) is added to `maxeventsperrow`.
  It is often the case that spurious events occur in nearby rows as well (in the same frame). This is addressed by checking rows within `widthnexttorow` of a bad one. As many events are rejected in those rows (by increasing distance to the bad row) as there are in excess of average.

- REJECT_E3 (IMAGING mode, not reduced IMAGING). Some instrumental effects result in wrong events associated with pathological (usually negative) charge values next to it. This information is known through the E_3 data. A simple way to reject this noise is to reject all events whose E_3 is less than a lower threshold (read from the E3THRES_ccd field in the `XMM_MISCDATA` CCF file). This is done with flag `UNDERSHOOT`, and has very little effect on true X-rays.

Sometimes the offsets applied by the EDU are much larger than what they should be for normal event detection. This can be intentional (to neutralise a noisy column, as is done on both sides of the window) but it can also happen serendipitously (bit flips in the EDU memory). REJECT_E3 looks for such large offsets (64 or more above normal) by looking for systematically low E_3 along particular rows or columns and solving the linear equation relating E_3 to the offsets. The result (as well as the normal patched columns) is written in the `OFFSETS` extension. The events directly on such rows or columns with wrong offset are flagged with `ON_BADOFFSET`. The test on E_3 is applied after correction for the additional offsets. This automatic detection of bad offsets is not applied in Small Window mode because the additional electronic noise in that mode confuses the algorithm. The detection may also be switched off manually (in other modes) via the `detectbadoffsets` parameter.

- REJECT_FLICKERING (IMAGING mode). In several CCDs, a pixel may suddenly release charge over several frames, more or less randomly over the CCD. This can result in 10 events or so very close to each other, and can lead to spurious source detection. Because those events follow each other very closely in frame, they may be detected by looking at peaks in `FRAME`, `RAWX`, `RAWY` space. They cannot be mistaken with a flaring source because they are much more spatially peaked than a source (most events are in the same pixel).
  Cosmic-rays can also result in peaks in FRAME, RAWX, RAWY space. Many, but not all, associated events have a "cosmic-ray" `PATTERN`. They are not as spatially concentrated as the flickering events, but they can be distinguished from a flaring source because they are usually very elongated (they are due to particles at grazing incidence).
  The search is done in boxes whose half-size in `FRAME`, `RAWX`, `RAWY` is defined by triplets provided by the user via `tolfxy`. The default is optimised for the flickering events. `tolfxy`="5 2 2 2 5 5" is better optimised for the cosmic-rays, but removes a small fraction of source counts (for that reason it is not default).

- SP_GATTI. In order to check the GATTI correction it is possible to compute the energy of events as it was seen by the ADC. This is done by adding the GATTI value and the offsets to all E_1 energies (the other E_i are sums and therefore unsuitable to this purpose) and forming their spectrum (for events unflagged by CUT_GTI). All small scale (in energy) structures in the charge to ADU conversion should reappear in this spectrum. After dividing it by the direct spectrum of E_1 energies convolved by the histogram of (GATTI + offsets) values, this can be compared with a reference spectrum to look for variations (outside **emevents**).
  In parallel, SP_GATTI has an active part, flagging for rejection with `REJECTED_BY_GATTI` all events with E_1 + GATTI + offsetx(RAWX) + offsety(RAWY) = 4095 (range of the ADC), which should have been rejected by the EMDH upper threshold but were not due to the GATTI mechanism.
  In TIMING mode this test is restricted to single events, as `ENGYE1E2` is the total charge over all pixels above threshold in the event. This routine is not applicable in Compressed TIMING mode (no `PATTERN`).

EV_REC, CUT_BAD, DIAGO and BINODAL may run only on a fresh ODF event file, not on the result of a previous run of emevents.
PUT_TIME, CUT_GTI and SP_GATTI require an associated frame file. If `frameset` is not set then an empty `EXPOSURE` extension is created to allow propagating the CCD-specific keywords after **evlistcomb**.

SP_GATTI and REJECT_E3 require an associated offset/variance file or extension. It is also advised to provide it for CUT_BAD and BINODAL. As it is not always obvious to know which offsets file is the right one (if several modes were used in an observation, there will also be several offsets files with different exposure numbers), **emevents** will select among a list of offsets files those compatible with the events file in terms of CCD, node, mode and window parameters. Among the compatible files **emevents** will select that with `DATE-OBS` closest to `DATE-OBS` of the events file, first within the offsets files obtained before the events file, then within those obtained after the events file if none was obtained before.
The first and last four column offsets (in the window) are patched to 4095 as is done in operations.
If no local offset/variance file is available (because the standard values were used), or if `offvarsets`is

not set **emevents** will read the offsets from the **cal**. If the offsets actually used were different, this will result in a large number of spGatti11 warnings and the flagging of events with truncated energy will be wrong.

3.1 Patterns

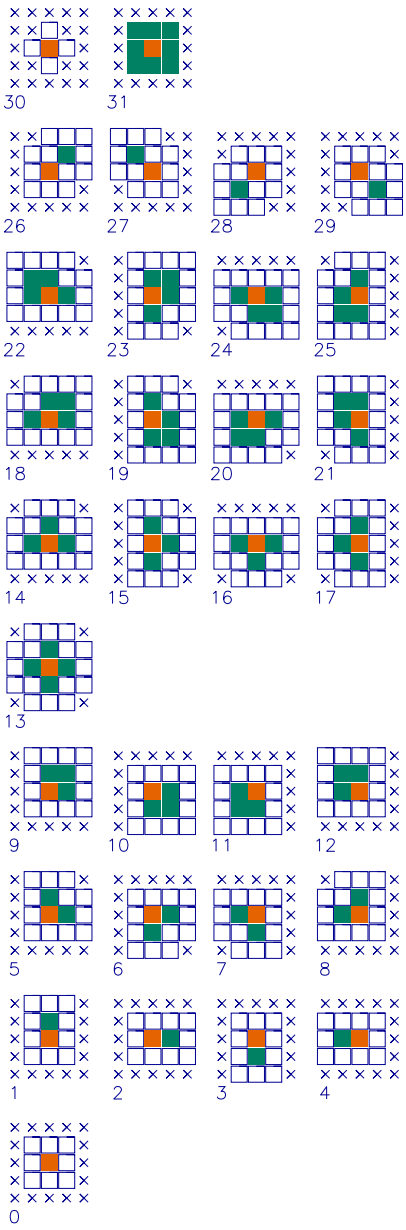

Figure 1: List of EPIC-MOS patterns (IMAGING mode)

Figure 1 for Imaging mode should be interpreted as follows:

- each pattern is included in a 5x5 matrix used for proximity analysis.

- a pattern is centered by definition on the pixel with highest charge.
- this central pixel is colored in red (its charge is E_1).
- the other pixels above threshold in the pattern are colored in green (the sum of their charges is E_2).
- all pixels colored in white must be below threshold.
- the crossed pixels are indifferent (they can be above threshold).
- the patterns are recognized in increasing order (this is important only for pattern 30, which comprises patterns 0 and 26-29).

The philosophy for patterns 0-25 is that a good X-ray pattern must be compact, with the highest charge at the center, and isolated (all pixels around are below threshold).

Patterns 26-29 are the so-called diagonal patterns, not expected from a genuine X-ray, but which can arise in case of Si fluorescence or of pileup of two monopixel events.

The E_3 and E_4 data in the ODF event lists have nothing to do with the "white" and "crossed" pixels.

- E_3 is the sum of charges of pixels below threshold in the 8 pixels of the 3x3 corona (their number is known from the pattern number).
- E_4 is the sum of charges of pixels below threshold in the 16 pixels of the 5x5 corona. Their number N4 is known from the `PERIPIX` column in the event list, which stores the number of pixels above threshold in the 5x5 corona. Thus N4 = 16 - `PERIPIX`.

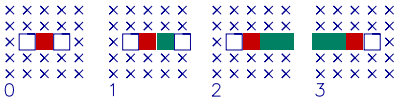

Figure 2: List of EPIC-MOS patterns (TIMING mode)

Figure 2 for Timing mode should be interpreted in the same way as in Imaging mode, with the difference that the place where maximum charge occurs is ignored. There are only crosses in the rows above and below that of the event. This means that the Timing pattern analysis is purely 1-D (it is insensitive to other rows), because each Timing 'row' is actually the sum of 100 true rows, so the rows are not physically related. Because there is no test for maximum charge, all doubles appear as `PATTERN`=1, whether leading or trailing.

Patterns 2 and 3 are not bounded. They are mostly not due to true X-rays, but to cosmic-ray tracks. After binning by 100 rows, cosmic-ray tracks appear as an horizontal string of pixels above threshold. Their beginning will be recorded by an event with `PATTERN`=2 and their end by an event with `PATTERN`=3. In other words, patterns 2 and 3 normally occur in pairs and define a horizontal string of pixels above threshold.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

odfeventset	yes	dataset	' '	none

Input events file

frameset	no	dataset	' '	none

Input frames file from emframes

newoutput	no	boolean	yes	yes/no

Create output events file (no for overwriting input)

eventset	no	dataset	'events.out'	none

Output events file. If this parameter is set, then `newoutput`=Y is automatic

offvarsets	no	list of datasets	' '	none

List of offset/variance files from ODF

othereventset	no	dataset	' '	none

Input events file for other node (2-node CCD read-out only)

otherframeset	no	dataset	' '	none

Input frames file for other node

analysepatterns	no	boolean	yes	yes/no

Use EV_REC subtask

flagbadpixels	no	boolean	yes	yes/no

Use CUT_BAD subtask

splitdiagonals	no	boolean	yes	yes/no

Use DIAGO subtask

keepsifluor	no	boolean	no	yes/no

Keep Si fluorescence diagonals as one

flagbadtimes	no	boolean	yes	yes/no

Use CUT_GTI subtask

setcoordinates	no	boolean	yes	yes/no

Use PUT_XY subtask

randomizeposition	no	boolean	yes	yes/no

Randomize `DETX`/`DETY` within a pixel

settimes	no	boolean	yes	yes/no

Use PUT_TI/PUT_TT subtask

randomizetime	no	boolean	no	yes/no

Randomize `TIME` within a frame

rejectrows	no	boolean	yes	yes/no

Reject rows with too many events

maxeventsperrow	no	integer	4	> 0

Maximum number of events per row in a frame

widthnexttorow	no	integer	10	≥ 0

Number of rows next to bad rows to flag as well

rejectbade3	no	boolean	yes	yes/no

Reject rows with too negative E3

detectbadoffsets	no	boolean	yes	yes/no

Automatically detect bad offsets in the data (not in SW mode)

rejectflickering	no	boolean	yes	yes/no

Reject flickering events

tolfxy	no	list of integers	5 2 2	≥ 0

Triplets of tolerances along FRAME, RAWX, RAWY

flagtruncatede1	no	boolean	yes	yes/no

Use SP_GATTI subtask

writegattispecset	no	boolean	no	yes/no

Create output GATTI spectrum file

gattispecset	no	dataset	'gattispec.out'	none

Output GATTI spectrum file

blocksize	no	integer	10000	> 0

Size of event blocks

5 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.

getParamValues03 *(error)*
output event file name = input events file name

openEventsIn01 *(error)*
`newoutput` = no (overwrite) and ODF-style events file

openEventsIn02 *(error)*
Not event list extension name in events file

openEventsIn03 *(error)*
Not EPIC MOS extension name in events file

openEvents2In01 *(error)*
invalid event file from other node

copyColumnPart01 *(error)*
column of unknown type in events file

getNvalid01 *(error)*
frame number in events file outside those in frames file

readFrames01 *(error)*
non-regular frame numbers in frames file

read2Frames01 *(error)*
frame numbers not parallel for both nodes

read2Frames03 *(error)*
invalid frame file from other node

initEdusoft01 *(error)*
unexpected `EDUTHR` in events file

initEdusoft02 *(error)*
unexpected `EDUMODE` in events file

initEdusoft03 *(error)*
Imaging mode data and `EDUMODE` not 3

initEdusoft04 *(error)*
Timing mode data and `EDUMODE` not 1

initEdusoft05 *(error)*
Reduced Imaging mode data and `EDUMODE` not 2

cutBad01 *(error)*
more than 2 events created out of one in CUT_BAD

addOffsets01 *(error)*
`flagtruncatede1`=Y but no offset/variance extension

getParamValues10 *(warning)*
`flagbadpixels`=Y but missing/invalid bad pixels extension in events file. Option is ignored
corrective action: run **badpix** on input event file

getParamValues11 *(warning)*
`splitdiagonals`=Y or `rejectbade3`=Y but data not in Imaging mode. Option is ignored
corrective action: none

getParamValues17 *(warning)*
`analysepatterns`=Y, `flagbadpixels`=Y, `splitdiagonals`=Y or `othereventset` is set but the input file already went through **emevents**. Option is ignored
corrective action: restart from the output of **emframes** and **badpix** if you wish to reapply EV_REC, CUT_BAD, DIAGO or BINODAL

getParamValues18 *(warning)*
`randomizetime`=Y but data in Timing mode. Option is ignored
corrective action: none

getParamValues19 *(warning)*
`analysepatterns`=Y, `flagbadpixels`=Y or `flagtruncatede1`=Y but data in Compressed Timing mode. Option is ignored
corrective action: none

getParamValues21 *(warning)*
`analysepatterns`=Y but `flagbadpixels`=N and `othereventset` is nt set. Option is ignored
corrective action: none

readFrames10 *(warning)*
no `FLAG` in frame file. Skip CUT_GTI
corrective action: check what went wrong in **emframes**

readFrames11 *(warning)*
CR_DEAD not run on frame file. Continue assuming no dead time.
corrective action: run **emframes** with `setdeadtime`=Y

readFrames12 *(warning)*
no `GATTIVAL` column in input frame file. Skip SP_GATTI
corrective action: run **emframes** with `setgatti`=Y

readFrames14 *(warning)*
The `CHECKFRA` keyword is not set in the frame file. Continue anyway
corrective action: check that the frame file was processed by **emframes**

readFrames15 *(warning)*
the input frame file cannot be used. CUT_GTI, PUT_TI/PUT_TT and SP_GATTI are skipped
corrective action: the input frame file should be the output of **emframes**. Check what went wrong in **emframes**

readFrames18 *(warning)*
`writegattispecset`=Y but GATTI off. Continue but no GATTI histogram will be produced
corrective action: none

read2Frames10 *(warning)*
no `FLAG` in secondary frame file. Continue assuming all frames from secondary node are valid
corrective action: check what went wrong in **emframes**

addOffsets10 *(warning)*
no offset/variance extension but `flagbadpixels`=Y, `rejectrows`=Y or `othereventset` is set. Assume constant and continue. This is non standard but has no major effect
corrective action: consider restarting from the output of **emframes** and **badpix**

checkOffsets10 *(warning)*
offset/variance file does not exist. Continue
corrective action: check this was not a typing error

checkOffsets11 *(warning)*
no valid offset/variance file but `flagbadpixels`=Y, `rejectrows`=Y, `flagtruncatede1`=Y or `othereventset` is set. Continue using offsets from CAL
corrective action: if you really need specific offsets, look for them in neighbouring ODFs

findBestTime12 *(warning)*
no compatible offset/variance file older than current exposure. Consider files obtained after current exposure
corrective action: check you have not forgotten an OVE.FIT file

checkRejected10 *(warning)*
more than 90% events rejected. Continue. This is unusual but may happen in normal conditions
corrective action: check the log (with verbosity set to 4 at least) to know why those events were rejected

reanalyze13 *(warning)*
no event survived CUT_BAD and EV_REC
corrective action: check nothing went wrong

reanalyze15 *(warning)*
`blocksize` smaller than number of events in a single frame. `blocksize` is increased internally
corrective action: this is often the sign that something is deeply wrong with at least one frame. check `NVALID` in the frame file and remove the events belonging to frames with very large `NVALID`

putXY10 *(warning)*
PUT_XY was already applied before. This changes the `RAWX` and `RAWY` columns and may not be applied twice. EV_REC, CUT_BAD and BINODAL may not be reapplied either
corrective action: restart from the output of **emframes** and **badpix** if you wish to reapply PUT_XY

rejectRows11 *(warning)*
`rejectrows`=Y but PUT_XY not applied. REJECT_ROWS not done
corrective action: run **emevents** with `setcoordinates`=Y

rejectRows12 *(warning)*
Exposure loss exceeds 1% at some rows (not corrected for)
corrective action: run **emevents** with -V 6 to know which

spGatti11 *(warning)*
`ENERGYE1` incoherent with GATTI and offsets. Flag event for rejection and continue. This may affect the flare screening light curve, and may also leave a bump in the spectrum above 12 keV
corrective action:

- if this warning is isolated, don't worry
- if this is an early observation (revolution 102 or before), or a mode other than full frame, then you probably need specific offsets. Restart **emevents** with `offvarsets` set to the list of all OVE.FIT files in the ODF
- if several happen in close sequence, it usually means that the frame number reconstruction in **emframes** failed at some point. The flare screening light curve will be wrong in that interval unless this is the central CCD
- if several happen along the whole exposure, but at the same X or Y, inform SOC (this is most likely a CCF error)

checkValid14 *(warning)*
invalid `RAWX`, `RAWY` or `PATTERN`. Reject that event. This can only be the result of an error on board, in the transmission or on the ground
corrective action: if several of those happen in close sequence, check ODF quality around those frames

initSifluor11 *(warning)*
something went wrong in the Si fluorescence recognition. This affects a very small fraction of events
corrective action: contact developer

readBadpix15 *(warning)*
a bad pixel has invalid coordinates. Ignore it and continue
corrective action: inform SOC (this is probably a CCF error)

openEventsIn12 *(warning)*
the events file is empty. Continue anyway (the exposure may be useful)
corrective action: check this is normal

openEventsIn14 *(warning)*
`offvarsets` was set but the input file already went through **emevents**. Option ignored
corrective action: restart from the output of **emframes** and **badpix** if you wish to change the offsets

getNvalid10 *(warning)*
events were found beyond the last frame in the frame file. They are rejected
corrective action: check the frame file was complete

getNvalid11 *(warning)*
an event was found with invalid frame number. It will be rejected
corrective action: check quality of neighbouring events as well

getXmmDate13 *(warning)*
invalid or absent `DATE-OBS` keyword in file. Continue setting the date to 01/01/2000. This is used only for ordering event and offsets files
corrective action: check `DATE-OBS` keyword

6 Input Files

1. EPIC MOS event list file for one CCD/node (from ODF/SDF, **emframes** or**emevents**) with bad pixels extension (from **badpix**).
2. frame file for that CCD/node (from **emframes**). uses columns `FRAME`, `TIME`, `FLAG`, `CRRATIO`, `GATTIVAL`, and keywords `TELESCOP`, `INSTRUME`, `OBS_ID`, `EXP_ID`, `CCDID`, `CCDNODE`, `GATTI_ON`, `SRC_RAWX`, `SRC_RAWY`.
3. EPIC MOS offset/variance file for that CCD/node (from ODF/SDF)
4. EPIC MOS event list file for the other node of the same CCD (from ODF/SDF) with bad pixels extension (from **badpix**).
5. frame file for the other node (from **emframes**)

The structure of files in the ODF is described in [1].

7 Output Files

1. new event list file with an `EVENTS` extension with a different number of rows and modified/additional items:
   - modified columns `RAWX`, `RAWY` (for PRODUCT: EPIC event list)
   - new INTEGER*2 columns `DETX`, `DETY` (0.05") (for **attcalc** and PRODUCT: EPIC event list)
   - new REAL*8 column `TIME` (s) (for PRODUCT: EPIC event list)
   - new INTEGER*2 column `FLAG` (for **emenergy**)
   - new REAL*4 columns `PHA` (ADU) and `PI` (eV) (for **emenergy**). In TIMING mode the original `ENGYE1E2` column is moved to `PHA`.
   - new empty INTEGER*4 columns `X`, `Y` (0.05") (for **attcalc**)
   - modified columns `ENERGYE1` and `ENERGYE2` (E_1 and E_2 in the text) These columns do not exist in TIMING mode.

- modified WINDOW keywords (for PRODUCT: EPIC event list)
- keyword DET_SYST (for **attcalc**)
- keywords RAND_XY and RAND_TIM telling which columns were randomised.
- keywords FRMTIME, GATTI_ON, COSMICSI, COSMICOU, and CRPATi propagated from the frame file
- keyword BINODAL warning of 2-node read-out (for **emenergy**)
- XMMEA_nn keywords detailing which flags were set, including XMMEA_EM setting the EPIC MOS rejection mask
- keywords detailing which subroutines were activated, in particular SETCOORD indicates that RAWX/RAWY were changed.
- comment lines with names of input files

a BADPIX extension propagated from the input file, an OFFSETS0 extension propagated from the offset/variance file (if any, for **emenergy**), an OFFSETS extension containing the list of bad offsets (for **emenergy** and **eexpmap**, **arfgen**, **lccorr**), with the following items:

- INTEGER*2 column RAWX giving the row (if OFFSETY¿0) or column (if OFFSETX¿0) of the bad offset
- INTEGER*2 column OFFSETX giving the amplitude of the additional column offset (0 for row offsets)
- INTEGER*2 column OFFSETY giving the amplitude of the additional row offset (0 for column offsets)

and an EXPOSURE extension specific to that CCD/node (for **evlistcomb** and **eexpmap**, **arfgen**, **lccorr**, [2]), with the following items:

- REAL*8 column TIME (s) giving the central time of each frame
- REAL*4 column TIMEDEL giving the integration time of each frame
- REAL*4 column FRACEXP giving the fraction of effective time

2. a GATTI spectrum file (for calibration purposes) with a SPECTRUM extension with ADU as index (1-4096) and the following items:
   - INTEGER*4 column ORIGINAL giving the histogram of E_1*GATTI
   - REAL*4 column SMOOTHED giving (normalised histogram of E_1) convolved with (histogram of GATTI)

8 Algorithm

```
MODULE em_events_module

   Read the file names of event files (input and output) and frame file

   Open files
   Get keyword values

   parameters read : choice of sub_tasks to execute
   Loop over input task parameters
      Read the task parameter / 1 to perform the procedure,
                                0 not to perform it  /
   End loop
```

```
   Get the values of files columns in memory

SUBROUTINE CUT_BAD
   if cut_bad requested  then
     read bad pixels in extension
     Loop over frames
       Creation of image using inverse pattern definition
       Connection with binodal task
          in the case of 2 files (prime and redundant) analysis
       Set bad pixels to 0
       Loop over events
          Use of EDUSOFT code for new pattern reconstruction
          creation of new rows and destruction of old ones
       end loop
     end loop
   endif

SUBROUTINE DIAGO
   if diago requested then
     Loop over event rows for diagonal patterns
       Check if both E1 and E2 energies are different from Si K energy
       In this case :
         first correct the existing pattern
         pattern is 0, e2 is 0, others the same
         second create another pattern
         pattern is 0, e1 is old e2, X & Y to be calculated
         depending on pattern
         e2 is 0, others the same
         both patterns are flagged
         ( do not correct the NVALID in the file frame )
       end loop
       new rows have been created at the end of file
       now shift of lines in order to insert new events
       Loop over number of new lines
          Shift all rows following place of insertion
       end loop
    endif

    Check events with energy value below EMDHLOW or above EMDHHIGH

SUBROUTINE CUT_GTI
    if cut_gti requested  then
      Loop over frame rows
         all events whose frames were flagged are flagged for rejection
      end loop
    endif

SUBROUTINE PUT_XY
    if put_xy requested then
      Loop over events
         center the events, substract the underscan, add 1.
         flag the events outside the field of view
      end loop
      Modify window keywords
```

```
      Compute camera coordinates
    endif

SUBROUTINE PUT_TI  ( Imaging mode )
    if put_ti requested then
      Loop over frame rows
         Loop over events rows
           calculate the average time for the frame
           copy this time in the time column of events file
         end loop
       end loop
     endif

SUBROUTINE PUT_TT  ( Timing mode )
     if put_tt requested then
       Loop over frame rows
         Loop over events rows
           calculate the new time as reference frame time plus Y offset
           copy this time in the time column of events file
         end loop
       end loop
     endif

SUBROUTINE REJECT_ROWS
    if reject_rows requested then
       Loop over frame rows
         Loop over events rows
            Build RAWY profile
         end loop
         Loop over events rows
            Flag events with RAWY profile larger than maxeventsperrow
            If 2 or more, flag events within +/- widthnexttorow
               (in the same frame) as well
         end loop
       end loop
    endif

SUBROUTINE REJECT_E3
    if reject_e3 requested then
      if detectbadoffsets and not SW mode then
         Build median E3 on each row and column
         Look for successive rows/columns with median E3 < -100
         Solve E3(i) = - 3*offset(i-1) - 2*offset(i) - 3*offset(i+1)
         to deduce which offset is wrong
         Flag events on rows/columns with wrong offset
      endif
      Read e3threshold from CAL
      Loop over events rows
         Flag events with pattern <= 4 and E3 < e3threshold
      end loop
    endif

For each triplet in tolfxy, call REJECT_FLICKERING
SUBROUTINE REJECT_FLICKERING(ftol,xtol,ytol)
    Ignore events OUT_OF_CCD_WINDOW, IN_BAD_FRAME, ON_BADROW, ON_BADPIX
```

```
    volref = (2*ftol+1) * (2*xtol+1) * (2*ytol+1)
    Get map of average event density, smoothed at xtol, ytol
    Get number of events per frame, smoothed at ftol
    Loop over events
       Count number of other events within ftol, xtol, ytol
       Estimate local average from larger of averages
          over FRAME (for sources) and RAWX,RAWY (for flares)
       If Poisson probability to get observed counts is > 1E-6, count as peak
    end loop
    Keep only local maxima among peaks
    Loop over peaks
       Identify events closer than 2*ftol in FRAME, 10*(xtol,ytol) in RAWX,RAWY
       If spatial distribution is compatible with a source
          (not too peaked, not too elongated) don't flag
       else flag events close to the peak, up to the point
          where the density meets the average
    end loop

SUBROUTINE SP_GATTI
    if sp_gatti requested   then
      Loop over events rows
        calculate the E1 spectrum, the (E1+gatti) spectrum
        and the gatti spectrum
      end loop
      Convolution of E1 and gatti spectra
    endif

    sequence of instructions for outevent1 creation

    closing files

end module
```

9 Comments

- The reanalysis of events in EV_REC, CUT_BAD and BINODAL calls a C routine (edusoft) which interfaces to the C routine (edu_simu, written by M. Lortholary of the EPIC/Saclay team) which simulates the EDU recognition.
- The search for new bad pixels in the data should be performed after **emevents** so that known bad pixels are removed and positions are correct.
- Instead of zeroeing bad pixels in CUT_BAD, it may be possible to subtract their average level (usually well defined), leaving only increased noise.
- In 2-node operation, the task must be run twice (once for each node). Thus the projection of events back to the CCD map in BINODAL is done twice. This is not expected to be a big overhead. The alternative would be to generate both output event lists in the same task.

10 Future developments

The following points are not considered final:

- To recompute properly the E_2 energy across nodes (in 2-node operation) the differential gain of the nodes should be accounted for.

References

[1] ESA. XMM Interface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM). Technical Report XMM-SOC-ICD-0004-SSD Issue 2.5, ESA/SSD, June 2000. Found at the URL: `ftp://astro.estec.esa.nl/pub/XMM/documents/odf_icd.ps.gz`.

[2] L.Angelini I.George. Specification of Physical Units within OGIP FITS files. Technical Report OGIP/93-001, NASA/GSC, May 1995.