espfilt

March 6, 2025

Abstract

The task **espfilt** generates soft proton contamination-filtered products, including event lists, images, and lightcurves from pipeline-processed event lists. **Note Well:** when this routine was first implemented under ESAS (not SAS) it included a large amount of filtering based on PATTERN and FLAG. This was carried over into SAS21. Now, in order to make this a more "general user" routine, that filtering has been completely removed.

1 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING, TIMING
EPIC PN	IMAGING, TIMING

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

The task **espfilt** applies one of two user-chosen methods for filtering an XMM-Newton Event List (MOS or PN) of cosmic soft proton events. The methods both utilize corner (unexposed, outside of FOV) sections of the CCDs. Both methods extract a lightcurve from both the FOV and the corners.

The *histogram* method fits a rough Gaussian to a histogram of the FOV lightcurve values. The peak of that Gaussian is taken as the nominal count rate. A user specified number of σ around the peak (default:2.5) are taken to be "flare-free" The program then creates a GTI file for those "flare-free" time intervals and then creates a filtered event file using that GTI.

The *ratio* method takes the ratio of counts per pixel of an annulus near the edge of the FOV (and away from any bright central source) to that of the unexposed corners. The time intervals with a ratio below a user-defined threshold (default: 1.2) are taken to be "flare-free". The program creates a GTI for the "flare-free" time intervals and then creates a filtered event file using that GTI.

In both methods, a new filtered event list, lightcurves of the FOV and corners, GTI FITS files and filtered images are created. A QDP plot of the lightcurves are created if one is using the histogram method.

3.1 Generalities

This task takes an existing eventlist produced by emchain, epchain, emproc, epproc, or the XMM Pipeline (PPS), applies one of the two selectable methods, produces a GTI file, then creates a) a filtered event list, b) filtered image, c) FOV lightcurve, and d) a 'Corners' (corner pixels of outer chips) lightcurve. A QDP plot of the count rate histogram is created by the histogram method only. The QDP file must be converted to postscript or other forms by the user.

3.2 Main loop

This task assumes that SAS tasks **cifbuild**, **odfingest** and one of the event-list producing EPIC MOS or PN tasks (**emchain**, **epchain**, **emproc**, **epproc**) have previously been run, or the PPS data is present.

An error will result if the odfingest ASCII output file or the cifbuild FITS file are not present and set using the proper SAS environment variables or command line options.

1. The task checks to see if event list is present and has the requisite keywords.
2. Define corner selection criteria for evselect calls.
3. Create allim and allev (full CCD) images and event lists.
4. Create corev, corim, corlc (corner event lists, images, lightcurves)
5. Create fov (Field of view only) lightcurves.
6. Depending on the method selected, the proper subroutine for processing via that method is envoked. For the Histogram method: Clean lightcurve in subroutine clean_lc.

   ```
   Clean the light curve by creating a histogram of rate values from
   the light curve, finding the most likely value, assuming that to be
   similar to the mean of the quiescent rate, then fitting a Gaussian
   (or poissonian) to a small window around that value in the histogram
   to determine the true mean and dispersion of the quiescent
   background rate.  Excludes time intervals with rate higher than a
   multiple of the dispersion above the mean quiescent background and
   excludes good regions shorter than some (currently hardcoded)
   limit.
   ```

   For the Ratio Method:

   (a) Create FOV annulus and corner annulus images and lightcurves.
   (b) Determine Counts per pixel Ratio between the FOV annulus and corners.
   (c) For Time Intervals within user-selected ratio threshold ranges, select data using evselect.
7. Create an ASCII QDP format file (histogram method only)
8. Run gtibuild.

9. Run evselect to produce filtered, flare-free events file.
10. Run evselect to produce filtered, flare-free image.
11. Run evselect to produce corner-only events file.
12. Run evselect to produce corner-only image.
13. If PN, run evselect to produce cleaned OOT event file.

3.3 Methods

Currently, only the Histogram and Ratio methods described above operate as required. The Histogram method effectively cuts the light curve at some number of sigma above the mean nominal level and the Ratio method compares the count-rate per unit area in the obscured corners of the detector to the same in a FOV annulus within the 8-12 keV band.

The task leaves open the possibility to add new "methods" in future releases.

3.4 Cleaning lightcurves

For the Histogram Method, clean_lc was originally a FORTRAN 77 program developed by Steve Snowden and K.D. Kuntz in 2005-6. The task espfilt was originally a perl script that operated on an input event list then used a system call to envoke clean_lc. The perl script and the F77 program have been converted entirely to FORTRAN 95, accessing XMM-Newton datasets via the DAL.

Clean_lc's purpose is to clean a light curve by creating a histogram of rate values from the light curve, finding the most likely value assuming that to be similar to the mean of the quiescent rate, then fitting a Gaussian to a small window around that value in the histogram to determine the true mean and dispersion of the quiescent background rate. The program then excludes time intervals with rate higher than a multiple of the dispersion above the mean quiescent background and excludes good regions shorter than some limit.

3.5 Examples

The following are several examples, with descriptions, of running espfilt:

- espfilt eventfile=mos1S001.fits

  ```
  Invoking the task with only an event file will cause it to
  default to binning the data into 60 second bins and smoothing
  it. The default min and max channels will be 2500eV and
  8500eV respectively.
  ```

- espfilt eventfile=mos1S001.fits withsmoothing=yes smooth=25

  ```
  This will change the default smoothing from 51 to 25.
  ```

- espfilt eventfile=mos2S002.fits elow=500 ehigh=8000

```
This command will change the default minimum and maximum energy ranges.
```

- espfilt eventfile=mos2S002.fits method=ratio ratio=1.5 withbinning=yes binsize=100

```
This command will run espfilt in ratio mode with a user selected
FOV to unexposed corner ratio of 1.5 instead of the default
1.2 and binning the data into 100 second time bins rather than
the default of 60.
```

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

General parameters

eventfile	yes		dataset	none

Input FITS event list

withoot	no	boolean	no	yes—no

Input a PN OOT event list? (only for PN)

ootfile	no		dataset	none

PN OOT FITS event list (only for PN)

method	no	string	histogram	ratio—histogram

fit method to use.

withsmoothing	no	boolean	yes	yes—no

Smooth data?

smooth	no	integer	51	$1 <= smooth < 60$

Smoothing factor in seconds

withbinning	no	boolean	yes	yes—no

Bin data?

binsize	no	integer	60	$1 < binsize < 100$

Bin width in seconds

ratio	no	real	1.2	$0 <= ratio <= 10.$

Flaring ratio of annulus counts

elow	no	integer	2500	$0 < elow <= 11999$

Low Band Energy

ehigh	no	integer	8000	$1 < ehigh <= 12000$

High Band Energy

rangescale	no	real	6.0	$5. < rangescale < 25.$

Histogram fit range scale factor

allowsigma	no	real	2.5	0.1 <= *allowsigma* <= 10.

Limit in sigma for unflared rates

limits	no	real	0.1,6.5	0.1 <= *limits* <= 10.

Parameter limits for gaussian fit (plim)

keepinterfiles	no	boolean	no	yes—no

Keep intermediary files produced?

NOTE: The *rangescale* parameter default is appropriate for the MOS, but not for the pn; for the pn the default should be 15.0

Parameters for individual tasks

Standard SAS parameters
Some SAS options are also interpreted by **espfilt** before being passed to its constituent tasks:

- '-V' (verbosity) is used in the same way as the SAS_VERBOSITY environment variable.
- '-o' (odf) is used to define the directory where the data resides in the same way as the SAS_ODF environment variable. Normally, the task checks the environment variable SAS_ODF, then checks if a valid **odfingest**-produced ASCII file is present, but the user can enter the odf directory or file on the command line with this option.
- '-i' (ccf) is used to define the directory where the Calibration (CCF) data resides in the same way as the SAS_CCF environment variable. Normally, the task checks the environment variable SAS_ODF, then checks if a valid **cifbuild**-produced ASCII file is present, but the user can enter the ccf directory or ccf.cif file on the command line with this option.
- There is emulation of the '-h' (help), '-m' (manpage), '-p' (param) parameters.

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.

badEnergyChans *(error)*
elow GT ehigh

noClosed *(error)*
Does not work with filter=Closed ODFs

noOOTwMOS *(error)*
Will not process PN OOT file for MOS input EVLI

noPNFast *(error)*
Espfilt does not support EPN Fast mode

noPNSmallWindow *(error)*
Espfilt does not support EPN Small Window mode

UnsupportedMethod *(error)*
Method must be either histogram or ratio

badInfile *(error)*
Input event list does not have necessary components.

noGTI *(error)*
GTI file does not exist

badInfile *(error)*
Problems reading, accessing table, columns in event file.

badTable *(error)*
Input or output file table missing critical Column

noCounts *(error)*
All histo counts are zero! Check your FOV Lightcurve!

PNLargeWindow *(warning)*
EPN PrimeLargeWindow SUBMODE may produce suspect results
corrective action: attempts analysis

RatioBinLT100 *(warning)*
Binsize for Ratio Method should be ¿= 100
corrective action: attempts analysis

noIntervals *(warning)*
Did not find any intervals that meet criteria
corrective action: attempts analysis

lowOutOfRange *(warning)*
low outside acceptable range
corrective action: sets L_IMME to -9.99E+99

MaxPLCltHigh *(warning)*
Max plc value outside acceptable range
corrective action: sets H_IMME to -9.99E+99

HfitEvalBlownUp *(warning)*
Components for calculating H_IMME/H_TOTL may have blown up
corrective action: suggest alternate input params

LfitEvalBlownUp *(warning)*
Components for calculating L_IMME/L_TOTL may have blown up
corrective action: suggest alternate input params

NoOpenQDPfile *(warning)*
Problem open/read/write output QDP file
corrective action: attempts write to LUN

SingularMatrix *(warning)*
Singular matrix solution
corrective action: No Count Rate Histogram Produced

nnLTzero *(warning)*
Close interval running problem. NN is LT 0?
corrective action: suggest alternate input params

BadFittingParam *(warning)*
Your choices for input params are causing gaussian fit to fail
corrective action: suggest alternate input params

6 Input Files

1. A filtered event list FITS file created by the XMM-Newton Pipeline (PPS) or reprocessing via one of the EPIC chain or proc tasks. These normally have the form P??????????????????EVLI????.FIT but this is not strictly necessary because the user may specify any valid event file using the `eventfile` parameter. In the examples, files such as "mos1S001.fits" are short form for these chain-produced long file name event lists.

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

7 Output Files

Common files to both methods have the same names.

Note well: In instances where there are similar filenames, but where one has a 'c', that indicates the file is *cleaned*, e.g., mos1S001-fovev.fits is an event list before being cleaned by espfilt and the resulting cleaned event list from it will be mos1S001-fovevc.fits.

1. Filtered event list for the entire detector x-allevc.fits
2. Filtered image file for the entire detector (no other selection) x-allimc.fits
3. Lightcurve file for the FOV x-fovlc.fits
4. UNfiltered event list for the corners x-corev.fits (intermediary)
5. Filtered event list for the corners x-corevc.fits
6. Filtered image file for the corners (no other selection) x-corimc.fits
7. Lightcurve file for the corners x-corlc.fits
8. Post-filtering Good Time Interval (GTI) file x-gti.fits
9. Unfiltered FOV annulus event file x-annev.fits (ratio method only)
10. Unfiltered FOV annulus spectrum x-ann.pi (ratio method only)
11. QDP text file of histograms and lightcurves x-hist.qdp

"x" has the form {inst}{expid}, such as mos1S001.

8 Intermediate Files

There is a parameter, *keepinterfiles* that controls whether products produced from evselect calls are kept (true) or deleted (false). The default is to delete these files.

1. Good Times Interval text file x-gti.txt
2. Unfiltered corner event list x-corev.fits
3. Unfiltered corner image file (no other selection) x-corim.fits
4. Unfiltered image file for the entire detector (no other selection) x-fovim.fits

9 Algorithm

```
Below is the algorithm for the CORNER METHOD
   subroutine espfilt
      Read parameters.
       Create these with evselect
        FOV image
        FOV lightcurve
        Corner event list
        Corner image
        Corner lightcurve
        if (method .eq. ratio)
         Create FOV annulus event list
         Create FOV annulus image
         Create FOV annulus lightcurve
        endif
       Call method
        if (method .eq. histogram) then
 call histogram_method
 subroutine histogram_method:
          Construct intermediary file names (QDP, GTI)
  call clean_lc (based on clean_rel.pro)
  subroutine clean_lc:
   extract columns TIME, RATE from FOV LC
   extract columns TIME, RATE from corners LC
   bin TIME, RATE columns from object LC
   bin TIME, RATE columns from corners LC
   smooth RATE column from object LC
   smooth RATE column from corners LC
           create histogram sortindex array
           fill sortindex as simple histogram of counts
           find peak of the distibution
           run downhill simplex multi-D minimization of
           gaussian fit function
           verify goodness of fit
           call write_qdp_gti to create output quicklooks
  end clean_lc
  call gauss_fit
```

```
        end histogram_method
elseif (method .eq. ratio) then
 call ratio_method
   subroutine ratio_method:
          extract corner spectrum
          backscale corner spectrum
          extract FOV annulus spectrum
          backscale FOV annulus spectrum
          read exposure, backscal, counts from Corner data.
          read exposure, backscal, counts from FOV Ann data.
  create ratio of FOV annulus to corner:
   ratio =  fovann_counts / corner_counts * \
            (corner_time * corner_area) / \
            (fovann_time * fovann_area)
            call clean_lc as above
  create ratio of FOV annulus to corner LC
  create GTI for time intervals when FOV\/corner
          ratio acceptable
          call write_qdp_gti to create output quicklooks
 end ratio_method
end call_method
        Create these with evselect:
         FOV filtered (flare-free) event list
         FOV filtered (flare-free) image
         Corner filtered (flare-free) event list
         Corner filtered (flare-free) image
        Add EHIGH, ELOW attributes to products
        Add fit parameters attributes to products:
         NORM, WIDTH, CENTER
         H_IMME, H_TOTL, L_IMME, L_TOTL
       end espfilt
subroutine write_qdp_fit
 open QDP output file
 write header, title information
 write out count rate histogram
  overlay selection and fit limits
 write out FOV lightcurve
  overlay GTIs
 write out Corner lightcurve
  overlay GTIs
 open ASCII gti.txt file
  write out GTIs to ASCII
 task gtibuild create GTI fits file
        end subroutine write_qdp_fit
```

10 Comments

The Ratio Method suffers from a speed difference with the Histogram Method due to the running of task backscale.

11 Future developments

- Test Snowden's Poissonian curve method.
- Allow user-specified corner and image boundary pixel definitions.
- Allow user-specified inputs to task evselect from command line.
- Use direct pgplot calls to produce PS output rather than QDP file.
- increase matrix efficiency using Mukai's Marquadt F90 subroutines.

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`.