espfilt

June 2, 2019

Abstract

The task **espfilt** generates soft proton contamination-filtered products, including event lists, images, and lightcurves from pipeline-processed event lists.

1   Instruments/Modes

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

2   Use

pipeline processing	yes
interactive analysis	yes

3   Description

The task **espfilt** applies one of (currently) 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 each of the FOV and corners. The first method fits a roughly Gaussian peak at the nominal count rate, creates a GTI for those time intervals with count rates within the thresholds. The second method takes the ratio of counts per pixel (in the 8-12keV range) of an annulus near the FOV (and away from any bright central source) to that of the unexposed corners. The user sets a threshold ratio (default: 1.2) and creates a GTI for those time intervals when the FOV is below the threshold.

In both methods, a new filtered event list, lightcurves of the FOV and corners, GTI FITS files, filtered images and a QDP plot of the lightcurves are created.

3.1   Generalities

This task takes an existing eventlist, applies one of two selectable methods, and produces a GTI file, then creates a filtered eventlist, filtered image, qdp plot of the count rate histogram (for now, the QDP file

must be converted to postscript, etc, by the user), FOV lightcurve and a 'Corners' lightcurve.

3.2 Main loop

The main routine accepts a space-separated list of event files of any combination of MOS and PN exposures.

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

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.

For the Corner Method, espfilt performs the following in sequence:

1. The task checks to see if cifbuild and odfingest has been run.

2. The task checks to see if event list(s) is/are present.

3. Depending on the method selected, the proper subroutine for processing via that method is envoked.

4. For the corner method:

    (a) Create Source Image using evselect.

    (b) Create Source Lightcurve using evselect.

    (c) Define corners (currently hardcoded, may become User Specifiable).

    (d) Create Corner Image using evselect.

    (e) Create Corner Lightcurve using evselect.

    (f) 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, or user defined curve) 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:

1. Create FOV annulus Image using evselect.

2. Create FOV annulus Lightcurve using evselect.

3. Define corners.

4. Create Corner Image using evselect.

5. Create Corner Lightcurve using evselect.

6. Determine Counts per pixel Ratio between the FOV annulus and corners.

7. For Time Intervals within user-selected ratio threshold ranges, select data using evselect.

For both methods, each of the following is performed

1. Create an ASCII QDP format file.

2. Run gtibuild.

3. Run evselect to produce filtered, flare-free events file.

4. Run evselect to produce filtered, flare-free image.

5. Run evselect to produce corner-only events file.

6. Run evselect to produce corner-only image.

3.3   Methods

Currently, only the Corner and Ratio methods described above operate as required. The Corner 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.

One future "method" may include one that determines a fixed count rate plus some "delta". In this case, the high energy count rate, say 8-12 keV, is created and the user selects an appropriate threshold. This will be the "delta method". This method should be able to select the light curve produced by the PPS which are point source excluded (*FBKTSR* files).

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

3.4   Cleaning lightcurves

For the Corner Method, Clean_lc was originally a FORTRAN 77 program developed by Steve Snowden and Kip Kuntz in 2005-6. The task espfilt was originally a perl script that operated on an input eventlist 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   Customisation

Command line options may be used to modulate certain parameters. These modifications include:

- `eventset` parameter, any valid MOS or PN eventlist. can also be a list of event lists. Espfilt will loop through each.

- The parameter `smooth` allows the user to select a smoothing factor (with default of 60).

- The parameter `binsize` allows the user to select a time bin width (in seconds) for the lightcurves created (default is 60)

- The parameters `specchanmin` and `specchanmax` allow a user to set upper and lower energy ranges, in eV (effectively channel number) paired with the parameter `withspecranges`

- The parameter `ratio` allows the user to select a threshold for the ratio of the counts per pixel in the 8-12keV range between a FOV annulus and the counts in unexposed corners (default is 1.2)

3.6   Examples

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

- espfilt eventset=mosM2S011.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 12000eV respectively.
  ```

- espfilt eventset="file1 file2 file3" withsmoothing=yes smooth=25

  ```
  this command will cause espfilt to loop through the 3 individual files
  listed, and change the default smoothing from 60 to 25.
  ```

- espfilt eventset="MOS2S002.fits" withspecranges=yes specminchan=500 specmaxchan=12000

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

- espfilt eventset="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 rat:
  ```

4   Parameters

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

Parameter	Mand	Type	Default	Constraints

General parameters

eventset	no	string		none

list of event files

method	no	string		ratio

which method to use.

withsmoothing	no	boolean	N	Y/N

Smooth data?

smooth	no	integer	50	> 1

Smoothing factor in seconds

withbinning	no	boolean	N	Y/N

Bin data?

binning	no	integer	50	> 1

Bin width in seconds

withspecranges	no	boolean	N	Y/N

Use upper/lower spec chans?

specchanmin	no	integer	2500	> 1 ev, < 32766

Low Spectral Channel

specchanmax	no	integer	12000	> 2 ev, < 32767

High Spectral Channel

ratio	no	real	1.2	> 0.01, < 10.0

Flaring ratio of annulus_cnts corn_area corn_cnst annu_area

clobber	no	boolean	yes	T/F

Clobber existing files?

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.

odf *(error)*
> SAS_ODF environment variable does not exist or improperly set.

odffiles *(error)*
> no valid file found for given specifications (data mode, exposure, etc.)

badsyntax *(error)*
> unrecognized or duplicate parameters.

parsing *(error)*
> Problems parsing event file.

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

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

evselect *(error)*
> Problems filtering one of the files with evselect.

nParams *(error)*
> Wrong number of parameters for Gaussian fit.

gaussian *(error)*
> Problem returning from Gaussian fit algorithm.

illegalModel *(error)*
> Illegal model type for Marquardt algorithm.

command *(error)*
> System error returned while invoking command.

noGTI *(error)*
> GTI intervals not produced or GTI fits file missing.

textwrite *(error)*
> Error writing to a text file, *.txt or *.qdp.

samefit *(warning)*

> *corrective action:* Gaussian fit the same as initial values.

6 Input Files

1. Filtered event list files created by the XMM- Newton Pipeline or reprocessing via one of the EPIC chain or proc tasks.PPS. These normally have the form P?????????????????EVLI????.FIT or 0100_0097820101_EPN_S005_ImagingEvts.ds, but this is not strictly necessary because the user may specify any valid event file using the **eventset** parameters, giving **espfilt** a list.

7 Output Files

Common files to both current methods have the same names.

1. Object filtered event list Pobsidinstexp-objevlifilt.FIT

2. Object filtered image file Pobsidinstexp-objimgfilt.FIT

3. Object lightcurve file Pobsidinstexp-objlc.FIT

4. Object 'raw' image file Pobsidinstexp-objimg.FIT

5. Corners filtered event list Pobsidinstexp-corevli.FIT

6. Corners filtered image file Pobsidinstexp-corimg.FIT

7. Corners lightcurve file Pobsidinstexp-corlc.FIT

8. Good Time Interval (GTI) file Pobsidinstexp-gti.FIT

9. FOV annulus event file Pobsidinstexp-fovevli.FIT

10. FOV annulus spectrum Pobsidinstexp-fovspec.FIT

11. Corner spectrum Pobsidinstexp-corspec.FIT

12. QDP text file of histograms and lightcurves Pobsidinstexp-hist.qdp

8 Intermediate Files

1. gti.txt (good times interval) text file of the form:

   ```
   P??????????I#S???-gti.txt
   where ????????? is the obsid, I is the instrument, # is the MOS instrument
   number (1 or 2), S can also be U for scheduled/unscheduled, and ??? is the
   exposure id.
   ```

2. various evselect files that are by default disgarded.

9 Algorithm

```
Below is the algorithm for the CORNER METHOD
   subroutine espfilt
      Read parameters.
       Call method
        if (method .eq. corner) then
 call corner_method
 subroutine corner_method:
  task evselect (to create object image)
  task evselect (to create object light curve)
  task evselect (to create corners image)
  task evselect (to create corners light curve)
  call clean_lc (to bin and smooth)
```

```
  subroutine clean_lc:
   extract columns TIME, RATE from object 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 from object
    find peak of the distibution
  end clean_lc
  call gauss_fit
  subroutine gauss_fit:
    find best paramater fit using Marquardt routines:
    MQM  maximum likelyhood fitting routine
    MRQ  compute the matrix elements for Marquard method
    GFUNCS calculate gaussian function distribution
    return best fit
   end gauss_fit
       end corner_method
elseif (method .eq. ratio) then
 call ratio_method
   subroutine ratio_method:
  task evselect (to create corners event list)
  task evselect (to extract corners spectral file)
  task backscale (to add area of corners region)
  task evselect (to create fov annulus event list)
  task evselect (to extract fov annulus spectral file)
  task backscale (to add area of fov annulus region)
  task evselect (to extract corners Lightcurve)
  task evselect (to extract fov annulus Lightcurve)
   extract time, counts, area for corners from spectrum and LC
   extract time, counts, area for fov annulus from spectrum and LC
    create ratio of FOV annulus to corner:
ratio =  fovann\_counts / corner\_counts *
        (corner\_time * corner\_area) / (fovann\_time * fovann\_area)
   extract columns TIME, RATE from object 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 from object
  call clean_lc
  create ratio of FOV annulus to corner LC
  create GTI for time intervals when FOV\/corner ratio acceptable
 end ratio_method
end call_method
call write_qdp_fit
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
 task evselect create object filtered event list
 task evselect create object filtered image
 task evselect create corners filtered event list
 task evselect create corners filtered image
Filtered event list, lightcurves, nice qdp plot are generated.
   end espfilt
```

10   Comments

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

11   Future developments

- Complete Delta methods.

- 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