The XMM SAS source detection tasks

Georg Lamer (Astrophysikalisches Institut Potsdam, glamer@aip.de)

October 08 2001

Abstract

This document describes the use of the XMM SAS source detection tasks. Section 1 gives a general overview of the detection package. In section 2 examples of various detection runs are given. In Section 3 common problems and their solutions are discussed.

1 Introduction

The XMM-Newton SAS package offers two methods of performing source detection on EPIC datasets.

1. The tasks forming the *edetect_chain* perform a two-stage sliding cell and maximum likelihood fitting detection (see section 1.1.
2. The task *ewavelet* provides the possibility to use a wavelet detection algorithm for EPIC source detection. (see section 1.2

1.1 Maximum likelihood fitting tasks

The following tasks are the constituents of the *edetect_chain*:

Task	Purpose	input data sets	output data sets
1. *eexpmap*	create exposure maps	images, attitude file	exposure maps
2. *emask*	create detection masks	exposure map	detection mask
3. *eboxdetect* (local mode)	sliding box detection	images exposure maps detection mask	local box list
4. *esplinemap*	create background maps	image, exposure map detection mask local box list	background map
5. *eboxdetect* (map mode)	box detect using bkg map	images, exp. maps detection mask background maps	map detect source list
6. *emldetect*	maximum likelihood fit	images, exp. maps detection mask background maps	final source list source map
7. *esensmap*	create sensitivity maps	exposure map detection mask background map	sensitivity map

The tasks 1.-7. can be started one after another from the command line or SAS GUI. Alternatively the task *edetect_chain* provides a more comfortable interface to these tasks.

The tasks *eboxdetect* and *emldetect* are able to work on a single image or simultaneously on a set of up to 54 images with equal coordinate systems. This feature can be used, for example, to run the tasks on the images of the 5 standard energy bands simultaneously. This method provides higher sensitivity than running detection on a single broad band image. The output lists then provide source parameters for each energy band as well as a summary of all bands including hardness ratios and total detection likelihoods. The images of the 3 different EPIC detectors can also be simultaneously processed using this feature.

1.2 Wavelet detection

The task *ewavelet* detects sources in an image using the mexican hat wavelet algorithm. The algorithm detects both point sources and extended sources in a single image. The task is easy to use and efficient, especially for the detection of extended sources. However, the resulting source parameters are generally less reliable than those provided by the *edetect_chain* tasks. *ewavlet* can also be used to generate background maps instead of using *eboxdetect* and *esplinemap*. Apart from an input image, only an exposure map as provided by task *eexpmap* is required to run *ewavelet*.

2 Data analysis recipes

2.1 Step by step use of edetect_chain tasks for point source detection

In this example we use the EPIC PN pipeline products to generate a source list based on the images in the 5 standard energy bands.

```
eexpmap attitudeset=P0110970101OBX000ATTTSR0000.FIT  \
        eventset=P0110970101PNS001PIEVLI0000.FIT \
        expimageset=''pnexp_1000.fits  pnexp_2000.fits pnexp_3000.fits
                      pnexp_4000.fits pnexp_5000.fits''  \
        imageset=''P0110970101PNS001IMAGE_1000.FIT'' \
        pimin=''200 500 2000 4500 7500'' pimax=''500 2000 4500 7500 12000''
```

Note how you can create exposure maps in several energy bands by providing a number of output file names and the corresponding PI channel ranges (pimin,pimax). You can only specify one input image though, it is only used as a coordinate reference.

```
emask expimageset=pnexp_2000.fits threshold=0.25 detmaskset=pn_mask.fits
```

The detector mask will be zero, where the exposure in pnexp_2000.fits is less than 25% of the maximum. Only one mask for each detector is needed.

```
eboxdetect usemap=no  likemin=8 withdetmask=yes detmasksets=pn_mask.fits \
           imagesets=''P0110970101PNS001IMAGE_1000.FIT P0110970101PNS001IMAGE_2000.FIT  \
                        P0110970101PNS001IMAGE_3000.FIT P0110970101PNS001IMAGE_4000.FIT \
                        P0110970101PNS001IMAGE_5000.FIT'' \
           pimin=''200 500 2000 4500 7500''  pimax=''500 2000 4500 7500 12000'' \
           expimagesets=''pnexp_1000.fits pnexp_2000.fits  pnexp_3000.fits
                          pnexp_4000.fits  pnexp_5000.fits'' \
           boxlistset=eboxlist_local.fits
```

Creating a source list using local background, likelihood threshold is 8.

```
esplinemap  bkgimageset=pnback_1000.fits  scut=0.005 nsplinenodes=16 \
            imageset=P0110970101PNS001IMAGE_1000.FIT expimageset=pnexp_1000.fit \
            withdetmask=yes detmaskset=pn_mask.fits withexpimage=yes

esplinemap  bkgimageset=pnback_2000.fits  scut=0.005 nsplinenodes=16 \
            imageset=P0110970101PNS001IMAGE_2000.FIT expimageset=pnexp_2000.fit \
            withdetmask=yes detmaskset=pn_mask.fits withexpimage=yes

esplinemap  bkgimageset=pnback_3000.fits  scut=0.005 nsplinenodes=16 \
            imageset=P0110970101PNS001IMAGE_3000.FIT expimageset=pnexp_3000.fit \
            withdetmask=yes detmaskset=pn_mask.fits withexpimage=yes
```

```
esplinemap  bkgimageset=pnback_4000.fits  scut=0.005 nsplinenodes=16 \
            imageset=P0110970101PNS001IMAGE_4000.FIT expimageset=pnexp_4000.fit \
            withdetmask=yes detmaskset=pn_mask.fits withexpimage=yes

esplinemap  bkgimageset=pnback_5000.fits  scut=0.005 nsplinenodes=16 \
            imageset=P0110970101PNS001IMAGE_5000.FIT expimageset=pnexp_4000.fit \
            withdetmask=yes detmaskset=pn_mask.fits withexpimage=yes
```

esplinemap uses the local box list to cut out sources (up to radius, where the upper limit of scut counts/per pixel is reached). The remaining image is the fit by a spline map. Note that we use fewer spline nodes for the 2 hard imgages, since they are likely to provide only poor count statistics for the spline fit.

```
eboxdetect usemap=yes  likemin=8 withdetmask=yes detmasksets=pn_mask.fits
           boxlistset=eboxlist_map.fits \
           imagesets=''P0110970101PNS001IMAGE_1000.FIT P0110970101PNS001IMAGE_2000.FIT \
           P0110970101PNS001IMAGE_3000.FIT P0110970101PNS001IMAGE_4000.FIT \
           P0110970101PNS001IMAGE_5000.FIT'' \
           pimin=''200 500 2000 4500 7500''  pimax=''500 2000 4500 7500 12000'' \
           expimagesets=''pnexp_1000.fits pnexp_2000.fits  pnexp_3000.fits
                          pnexp_4000.fits  pnexp_5000.fits'' \
           bkgimagesets=''pnback_1000.fits pnback_2000.fits pnback_3000.fits \
                          pnback_4000.fits pnback_5000.fits''
```

This creates a source list using background values from the background maps. This list is used as input for *emldetect*. The likelihood threshold is chosen lower than for *emldetect*, since the final selection of significant sources should be done by *emldetect*.

```
emldetect imagesets=''P0110970101PNS001IMAGE_1000.FIT P0110970101PNS001IMAGE_2000.FIT \
                      P0110970101PNS001IMAGE_3000.FIT P0110970101PNS001IMAGE_4000.FIT \
                      P0110970101PNS001IMAGE_5000.FIT'' \
           bkgimagesets=''pnback_1000.fits pnback_2000.fits pnback_3000.fits \
                          pnback_4000.fits pnback_5000.fits'' \
           expimagesets=''pnexp_1000.fits pnexp_2000.fits  pnexp_3000.fits  \
                          pnexp_4000.fits  pnexp_5000.fits'' \
           pimin=''200 500 2000 4500 7500''  pimax=''500 2000 4500 7500 12000'' \
           ecf=''10.596 6.8157 2.0542 0.99483 0.25933'' \
           mllistset=emllist.fits  mlmin=10 determineerrors=yes
```

Now we have the final source list. If energy conversion factors are supplied, count rates will be converted to correct flux values. The ECFs depend on filter and pattern selection.

```
srcdisplay imageset=P0110970101PNS001IMAGE_8000.FIT boxlistset=emllist.fits
```

Plots source positions from source list onto the broad band image.

Note: From v1.1 task srcdisplay *supports source lists from task* emldetect

2.2 Using the *edetect_chain* task

The Perl-script based task *edetect_chain* combines the above detection tasks in a more user friendly way.

In the following example we use *edetect_chain* to reproduce the analysis described in section 2.1.

```
edetect_chain attitudeset=P0110970101OBX000ATTTSR0000.FIT \
              eventsets=P0110970101PNS001PIEVLI0000.FIT \
              eml_list=pn_emllist.fits
              eml_determineerrors=yes \
              esp_nsplinenodes=16  \
              ecf=''10.596 6.8157 2.0542 0.99483 0.25933'' \
              pimin=''200 500 2000 4500 7500''  \
              pimax=''500 2000 4500 7500 12000'' \
              imagesets=''P0110970101PNS001IMAGE_1000.FIT \
                          P0110970101PNS001IMAGE_2000.FIT \
                          P0110970101PNS001IMAGE_3000.FIT \
                          P0110970101PNS001IMAGE_4000.FIT \
                          P0110970101PNS001IMAGE_5000.FIT''

srcdisplay imageset=P0110970101PNS001IMAGE_8000.FIT boxlistset=pn_emllist.fits
```

This does exactly the same as the procedure from section ?? with the only difference that we can only choose one number of spline nodes (`esp_nsplinenodes`) for all images. Although *edetect_chain* has a large set of parameters in order to control most of the individual task parameters, most of them can be left at there default values for most purposes.

2.3 Searching for extended sources

```
edetect_chain imagesets=''P0110970101PNS001IMAGE_2000.FIT P0110970101PNS001IMAGE_3000.FIT'' \
              eventsets=P0110970101PNS001PIEVLI0000.FIT \
              attitudeset=P0110970101OBX000ATTTSR0000.FIT \
              eml_list=pn_emllist_ext.fits \
              eml_fitextent=yes \
              pimin=''500 2000'' \
              pimax=''2000 4500'' \
              esp_nsplinenodes=12 \
              eml_fitextent=yes \
              eml_ecut=0.95 \
              eml_withsourcemap=yes

srcdisplay imageset=P0110970101PNS001IMAGE_2000.FIT boxlistset=pn_emllist_ext.fits
```

In this example the extent fitting functionality of *emldetect* is switched on. Since *emldetect* is very slow, when using a large number of images for extent fitting, we use only 2 energy bands. The parameter `eml_ecut` controls the radius around each source, that is used for fitting (in units of encircled energy of the PSF). It is here increased from 0.68 (default) to 0.95 . Reducing the number of spline nodes for the background map to 12 prevents *esplinemap* to model faint extend sources as a background feature.

The resulting source list contains the columns `EXT` (best fit σ in image pixels), `EXT_ERR`, and `EXT_ML`, a measure of the likelihood of source extent. Creation of sourcemaps with `eml_withsourcemap=yes` is also useful to recognize the detection of extended sources.

2.4 Wavelet detection: ewavelet

```
ewavelet expmapset=P0110970101PNS001EXPMAP8000.FIT \
         imageset=P0110970101PNS001IMAGE_8000.FIT \
         minscale=1.4 maxscale=16. threshold=4.\
         makerecon=yes recimageset=pn_reconstructed.fits \
         savebkgrnd=yes bkgrndset=pn_ewave_bkg.fits \
         srclistset=ewavelist.fits

srcdisplay imageset=P0110970101PNS001IMAGE_2000.FIT boxlistset=ewavelist.fits
```

Since *ewavelet* can only process one image at time, the broad band image is used. The detection threshold is set to 4 σ and the scale sizes of the mexican hat wavelets range from 1.4 to 16 image pixels. A reconstructed image and background image are saved.

Note: task srcdisplay *supports source lists from task* ewavelet *only from version 1.2 on.*

3 Trouble shooting

1. Problem: Localized Clustering of spurious sources.

Reason: Background map locally underestimates true background.

Solutions: If the presence of extended sources results in a poor spline fit, the quality of the fit can be improved by increasing the number of spline nodes (parameter `nsplinenodes` of task *esplinemap*) In some cases low background in short exposures or hard energy band images leads to poor spline fits, in this case the value of `nsplinenodes` should be decreased in order to improve the fit statistics.

2. Problem: Many spurious source detections due to warm pixels.

Reason: Warm pixels can be a problem in EPIC MOS images produced by earlier versions of the pipeline.

Solutions: The task *emldetect* offers a hot pixel filtering algorithm (Set `withhotpixelfilter = yes`). The problem of the warm pixels itself can be solved by creating a new event file from the ODF with the latest version of *embadpixfind*.

3. Problem: Many spurious source detections in soft EPIC PN images.

Reason: Soft band EPIC PN images show increased background and noise near the edges of the detector. These features often lead to spurious detections.

Solutions: The noise can be significantly reduced by selecting only single and double events for the images ("PATTERN<=12").

4. Problem: Significant sources missing from *emldetect* source list.

Reasons: Bad background map or detection threshold too high.

Solutions: In case the background map shows suspicous features, try the solutions to problem 1. If the detection threshold seems too high, check whether *eboxdetect* or only *emldetect* has missed the sources and change the respective detection threshold.

References