especget

November 4, 2014

Abstract

A procedure to generate source and background spectra and the resultant ARF and RMF files for spectral fitting.

1 Instruments/Modes

Instrument	Mode
EPIC	IMAGING, TIMING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

especget is a one-stop task which produces all the files necessary for the spectral fitting of an XMM source. The task allows the user to produce a source spectrum, background spectrum and the OGIP-compliant (spectral fitting within a single procedure.

The essential input parameters are the name of the XMM events table and a selection expression for the source and background areas, e.g. `srcexp`=”((DETX,DETY) IN circle(572,-635,1340))” `backexp`=”((DETX,DETY) IN circle(2572,-1635,1340))”. The minimum selection required in the source and background expressions is the spatial region, which may be specified in sky (X/Y) or detector coordinates. Other selections are defaulted as shown in Table 1. These defaults may be further restricted by including additional selections for these quantities in the input expressions, e.g. `srcexp="((DETX,DETY) IN circle(572,-635,1340))&& PATTERN==0"`. **NB:** The source and background expressions should use the same pattern and flag selections. The expressions must be compliant with the **selectlib** standard.

The metatask **especget** runs the tasks **evselect**, **arfgen** and **rmfgen**. It also calculates the size of the source and background areas, a job normally performed by the task **backscale** but here achieved by calling **arfgen** directly. For more details of the processing performed by these tasks please see their respective user documentation.

The end result is a set of files which can be used directly in a spectral fitting program (see section 3.3).

Table 1: **Default event selections**

Parameter	Camera	
	MOS	PN
PATTERN	0–12	0–4
FLAG	XMMEA_EM[a]	0

[a] The selection #XMMEA_EM represents a conglomerate of flags; see **evatt**.

3.1 RMF options

The steps to create the spectra, calculate the box areas and create the ARF are quick but the creation of an RMF may take more than half an hour to complete. To shorten the execution time a canned matrix may be downloaded from the SOC web-site (http://xmm.vilspa.esa.es/external/xmm_sw_cal/calib/index.shtm) and used in the processing by specifying parameters of the form:

`withrmfset`=yes `rmfset`=m1_r6_all_15.rmf

Advice on canned matrices is available from the SOC web-site or the document XMM-SOC-PS-TN-0043

If `withrmfset`=yes and `rmfset` is not set then the task will identify the name of the canned matrix which best suits the created source spectrum and write this name into the RESPFILE keyword in the header of the source spectrum (see 5.3).

If `withrmfset`=no then rmfgen is executed to produce a specific RMF for the source. The `extendedsource` option tells **rmfgen** to produce a response function which is correct for an extended source, i.e. one which has been averaged over the spatial extraction region assuming a flat source flux distribution.

3.2 ARF options

By default, an ARF will be created which has been corrected for the area lost by the gaps between CCDs and by bad pixels. This latter correction relies on a list of bad pixels being present in the header of the eventset specified in the `table` parameter. These corrections may be turned off by specifying `withbadpixcorr`=no. The ARF generation stage can itself be turned off by setting `witharfset`=no.

If the spatial selection has been specified in sky coordinates (X/Y) then **arfgen** needs to internally convert these into detector coordinates. The default conversion uses attitude information taken from the header of the newly created spectrum. On rare occasions this can cause a problem and so the option exists, (`useodfatt`=yes), to use the raw attitude files stored in the ODF directory pointed to by the environment parameter **SAS_ODF**.

The `extendedsource` option tells **arfgen** to produce the ARF relevant for an extended source. The distinction between the ARF needed for a point-like and extended source is rather subtle. In practice, for an extended source the ARF has no correction for encircled energy, the mirror vignetting is averaged over the whole source area and area lost due to bad pixels and chip gaps is not convolved with the Point Spread Function (PSF). All of these effects are toggled by the `extendedsource` parameter.

To calculate the effective area curve **arfgen** averages over a grid known as a detector map, which by default has five by five elements. In certain cases, such as a narrow annulus or large complicated region, this is insufficient to sample the region and can result in a null ARF or BACKSCAL value being returned. To avoid this the number of bins may be specified on the command line using the parameters `detxbins` and `detybins`.

See the **arfgen** documentation for further details.

3.3 Header keywords

The task writes the names of the created files into the source spectrum header keywords, BACKFILE, RESPFILE, ANCRFILE. These may be automatically read by spectral fitting programs to link the files and perform area weighted background subtraction, e.g. for `xspec`.

```
XSPEC>data myagn_src.ds
 Net count rate (cts/s) for file   1   5.049    +/-  2.6091E-02( 98.5% total)
   using response (RMF) file...       myagn_src.rmf
   using auxiliary (ARF) file...      myagn_src.arf
   using background file...           myagn_bgd.ds
   1 data set is in use
XSPEC>
```

3.4 Examples

3.4.1 A point source

```
especget filestem=myagn srcexp="((DETX,DETY) IN circle(572,-635,800))"
backexp="((DETX,DETY) IN annulus(572,-635,1600,3200))"
table=P0129360201M1S004MIEVLI0000.FIT
```

This command line produces the files `myagn_src.ds`, `myagn_bgd.ds`, `myagn_src.arf`, `myagn_src.rmf`.

The command will take quite a while to run because of the RMF generation stage . A PN observation processed with this command will take roughly twice as long as a MOS observation.

3.4.2 Setting the names of output files individually

It is possible to set the names of the output files separately without using the `filestem` keyword by:

```
especget srcspecset=agrb.pi bckspecset=cleansky.pi srcarfset=agrb_arf.ds srcrmfset=agrb_rmf.ds
srcexp="((X,Y) IN circle(25572,-24635,1200))"
backexp="((X,Y) IN circle(23000,-28000,1340))"
table=P0129360201PNS020PIEVLI0000.FIT
```

3.4.3 Using a pre-prepared matrix

```
especget filestem=myagn srcexp="((X,Y) IN circle(25572,-24635,1200))"
backexp="((X,Y) IN circle(23000,-28000,1340))"
table=P0129360201PNS020PIEVLI0000.FIT withrmfset=yes
rmfset=epn_ef20_sdY9.rmf
```

This command tells the task that the canned matrix *epn_ef20_sdY9.rmf* will be used for spectral fitting. The task should finish within a few minutes. NB: If `rmfset` is not defined then **especget** will set the name to that of the relevant canned matrix.

3.4.4 Defining the patterns and flags to use

```
especget filestem=myagn withrmfset=yes rmfset=epn_ef20_sdY9.rmf
srcexp="FLAG==0&&PATTERN==0&&((DETX,DETY) IN circle(572,-635,1340))"
backexp="FLAG==0&&PATTERN==0&&((DETX,DETY) IN annulus(572,-635,1600,3000))"
table=P0129360201M1S004MIEVLI0000.FIT:EVENTS
```

This will create source and background spectra from events with a pattern of zero and having `FLAG=0`. Restricting the pattern range to 0 can be useful when the source is piled up.

3.4.5 Analysing an extended source

```
especget filestem=myagn srcexp="((X,Y) IN circle(25572,24635,4340))"
backexp="((DETX,DETY) IN annulus(25572,24635,4340,6500))"
table=P0129360201M1S004MIEVLI0000.FIT:EVENTS extendedsource=yes
```

The `extendedsource` option tells **arfgen** to produce the ARF relevant for an extended source.

4 Final preparation for spectral fits

If the source has sufficient counts it should be rebinned to allow the gaussian statistics used in fitting software to be applied correctly. An example using the `GRPPHA` program could be:

```
> grppha myagn_src.ds myagn_src_grp.ds
GRPPHA[] group min 20
GRPPHA[] exit
 ...... exiting, changes written to file : myagn_src_grp.ds
 ** grppha 2.9.0 completed successfully
>
```

5 Parameters

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

Parameter	Mand	Type	Default	Constraints

withfilestem	no	boolean	yes	

Whether to specify the output file names using the `filestem` parameter.

filestem	no	string	myfiles	

Stem for the output filenames. A value ”myfiles” will produce the files, *myfiles_src.ds, myfiles_bgd.ds,* *myfiles_src.arf, myfiles_src.rmf*

srcspecset	no	string	srcspec.ds	

Name of the output source spectrum. This is only used if `withfilestem` is false in which case it becomes mandatory.

bckspecset	no	string	bckspec.ds	

Name of the output background spectrum. Only used if `withfilestem` is false in which case it becomes mandatory.

srcarfset	no	string	srcspec.arf	

Name of the output effective area file (ARF). This is only used if `withfilestem` is false and `witharfset` is true.

srcrmfset	no	string	srcspec.rmf	

Name of the output redistribution matrix (RMF). This is only used if `withfilestem` is false and `withrmfset` is false.

srcexp	yes	string		

Expression for extracting the source events.

backexp	yes	string		

Expression for extracting the background events.

table	yes	table specifier		event list table specifier

A table specifier which must point to an event list table in a data set. It must be in either of the forms `setname` or `setname:tableid` where `setname` must be the name of an existing data set and `tableid` the name of a table in the specified data set. If the first form, `setname`, is used, the event data are sought in the *first* block of the named data set.

withbadpixcorr	no	boolean	yes	

Whether to use bad pixels and chip gaps in the ARF calculation.

useodfatt	no	boolean	no	

Whether to use the ODF attitude file to construct position info when calculating the ARF.

extendedsource	no	boolean	no	

If set **true** then **arfgen** is run with parameters which are relevant to an extended source.

detxbins	no	int	5	

The number of x bins in the detector map.

detybins	no	int	5	

The number of y bins in the detector map.

witharfset	no	boolean	yes	

If set **false** an ARF will not be produced by the task and the geometrical area of the source and background regions will not be calculated and written into the BACKSCAL keyword.

withrmfset	no	boolean	no	

If set **false** an RMF is produced by running **rmfgen**. If **true** the RMF is taken from the file specified in the parameter `rmfset`.

rmfset	no	string		

The name of the RMF to be used when **rmfgen** is not to be run. If this parameter is not set and withrmfset is **true** then the task sets the name of the RMF to be that of the appropriate canned matrix.

withenergybins	no	boolean	false	none

If true use energy grid specified by energymin, energymax, and nenergybins, otherwise, use the grid defined in the CAL.

energymin	no	real	0	none

Used if withenergybins = true. Lower energy bound of matrix, in keV.

energymax	no	real	15	none

Used if withenergybins = true. Upper energy bound of matrix, in keV.

nenergybins	no	integer	30	none

Used if withenergybins = true. The number of bins in the energy grid = number of rows in RMF matrix.

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.

InvalidExpression *(error)*
The input expression for this spatial region was not valid.

invalidInstrument *(error)*
The Epic camera description contained in the INSTRUME keyword isn't recognised

NoEventsTable *(error)*
The input table didn't contain an EVENTS extension

7 Input Files

- an EPIC calibrated events list from the pipeline, the e*proc or e*chain tasks.
- Optionally, an EPIC RMF file, e.g. epn_fs20_sY0.rmf.

8 Output Files

- A source spectrum of name filestem_src.ds or srcspecset with the BACKSCAL, RESPFILE, ANCRFILE and BACKFILE keywords set
- A background spectrum of name filestem_bgd.ds or bckspecset with the BACKSCAL keyword set
- An ARF of name filestem_src.arf or srcarfset
- Optionally, an RMF of name filestem_src.rmf or srcrmfset

9 Algorithm

```
Set output filenames: srcspec, bckspec, arfset, rmf_set
             depending on values of <withfilestem> and related params.

Create a source spectrum

   expression = <srcexp> + PATTERN + FLAG selections
   evselect table=<table> expression=<expression>
            withspectrumset=yes spectrumset=srcspec

Create a background spectrum

   expression = <backexp> + PATTERN + FLAG selections
   evselect table=<table> expression=<expression>......
            withspectrumset=yes spectrumset=bckspec

Create an ARF for the source spectrum, if requested, and calculate
the source box area

   if <witharfset> {

     arfgen spectrumset=<srcspec> arfset=<arfset> withrmfset=<withrmfset>
          rmfset=<rmfset> extendedsource=<extendedsource>
          withbadpixcorr=<withbadpixcorr> badpixlocation=<table>
          setbackscale=yes keeparfset=<witharfset> useodfatt=<useodfatt>

Use arfgen to calculate the background box area

     arfgen spectrumset=<backspec> arfset=temparf.arf setbackscale=yes
          keeparfset=no withbadpixcorr=<withbadpixcorr>
          badpixlocation=<table> useodfatt=<useodfatt>

   }

Create an RMF if not supplied
   if (! <withrmfset>) {
      rmfgen spectrumset=<srcspec> rmfset=rmf_set
            extendedsource=<extendedsource>
   }
   else if (<rmfset> has not been set) {
      rmfset = name of relevant canned matrix
   }

Set the values of BACKFILE, RESPFILE and ANCRFILE in header of srcspec.
```

10 Comments

This task will benefit from any reduction in the execution time of rmfgen.

References