



# epnoise

January 12, 2017

## Abstract

Algorithm to reject soft X-ray noise in the EPIC-pn camera

## 1 Instruments/Modes

Instrument	Mode
EPIC PN	IMAGING

## 2 Use

pipeline processing	yes
interactive analysis	yes

## 3 Description

### 3.1 General

**epnoise** task removes soft X-ray noisy frames from EPIC-pn camera.

The task calculates the number of events per frame between 20 to 30 adu and removes those frames above a certain threshold defined by **noisecut** parameter. Once the noisy frames have been removed, the exposure time is updated accordingly.

To do this filtering, the **epnoise** task logic has been divided in two different steps.

During the first step, **epnoise** is run using as input the output files of **epframes** and **badpixfind** tasks. Then, **epnoise** identify the noisy frames, creates or updates the column NEVT\_FRM, containing the number of events per frame and CCD with PHA values below a certain threshold, and write keywords containing suggestions for subsequent filtering. Where the keywords are:



- LAMBDA: lambda of poissonian fit
- NORM: normalization of poissonian fit
- NEVT\_CUT:suggested cut value ("10
- NEVT\_ALT: alternative cut value (1.0)

Pixels which are affected by bright celestial sources in this energy range are removed through a mask generation. To create this mask for removing bright sources, epnoise calculate the median of the full image and apply a cut using the **sigmacut** parameter. Then a mask for badpixel is created and added to the previous mask. The **savemasks** parameter writes to disk the masks of all CCDs.

After this first step of **epnoise**, the rest of the EPIC-pn processing chain is executed, propagating the new column (NEVT\_FRM) and the new keywords (LAMBDA,NORM,NEVT\_CUT,NEVT\_ALT).

During the second step, **epnoise** filter the final event list using the NEVT\_CUT threshold for each CCD. The **epnoise** filter the final event list cretaing the following expression per CCD (NEVT\_FRM >= NEVT\_CUT). Update the STDGTI extension adding the gaps corresponding to the frames that have been removed. Then, update the ONTIME and LIVETIME keywords.

## 4 Parameters

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

Parameter	Mand	Type	Default	Constraints
-----------	------	------	---------	-------------

<b>set</b>	yes	filename		
------------	-----	----------	--	--

Name of the epframes output file

<b>eventSet</b>	no	filename		
-----------------	----	----------	--	--

Name of the calibrated event file

<b>identifynoisyframes</b>	no	boolean	yes	yes—no
----------------------------	----	---------	-----	--------

Identify Noisy Frames?

<b>applyfilter</b>	no	boolean	no	yes—no
--------------------	----	---------	----	--------

Keep output of filtering process?

<b>sigmacut</b>	no	real	3.0	
-----------------	----	------	-----	--

Sigma cut for bright sources

<b>noisecut</b>	no	int	2	>0
-----------------	----	-----	---	----

Noise cut (maximum allowed number of soft events in frame)



<b>savemasks</b>	no	boolean	no	yes—no
------------------	----	---------	----	--------

Save CCDs mask to a file

## 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.

### **MissingParameter** (*error*)

Missing input file name

### **TooManyFrames** (*error*)

Too many frames while computingcounting the events per frames.

### **RawEventFileEmpty** (*error*)

epframes output file is empty

### **NoisyEventsEmpty** (*warning*)

None noisy events filtered.

*corrective action:* Check the NEVT\_CUT value.

## 6 Input Files

1. The output file of **epframes** + **badpixfind** tasks (step 1).
2. The previous files (one for each CCD) + filtered event list (step 2).

## 7 Output Files

1. Event file with soft X-ray noisy filtered.

## 8 Algorithm

```
do i = 1,nfiles
  call identifynoisframes [step1]
  open file
  call framecounter
  call ftpois
  call dpoiss
  call factrl
  call gammln
```



```
call epnoisemask
  call createNoisyMask
call framecounter
call ftpois
  call dpoiss
    call factrl
      call gammln
close file
call writeInfo
call removenoisyyframes [step2]
enddo
call filterEventfile [step2]
```

## 9 Comments

- 

## References