



# epatplot

February 1, 2016

## Abstract

Plot EPIC pn and EPIC MOS event pattern statistics

## 1 Instruments/Modes

Instrument	Mode
EPIC PN	all (IMAGING, TIMING, BURST)
EPIC MOS	all (IMAGING, TIMING)

## 2 Use

pipeline processing	no
interactive analysis	yes

## 3 Description

The task reads the event pattern information of an input EPIC pn or EPIC MOS event set<sup>1</sup>.

It creates a plot with these distributions and fractions as function of PI channel. The fraction plot also contains model curves for single and double fraction for all modes (FF, eFF, SW, LW, TI, BU) as far as they are included in the QUANTUMEF CCF files.

The models for the single and double event fractions were derived to correct for corresponding event selections which were applied in accumulating an EPIC pn spectrum. The event fractions are included in the detector response matrices. To use the standard response matrix one needs to make sure that the pattern fractions of the events in the area selected for the spectrum follow the model. The EPIC MOS model curves were provided by the hardware group at LUX.

<sup>1</sup>e.g., event set with source region selected via `evselect`; if parameter `withbackgroundset=Y` (the default is “N”) an additional background event set is read in, and background subtracted pattern fractions are determined. Use parameter `backscal` to set the background-to-source extraction area ratio (default: 1.0). If exposure times of the source and background event set differ, the `backscal` parameter needs to be adjusted accordingly.



The single and double pattern fractions are highly sensitive to pile-up effects<sup>2</sup> in *bright sources* and in case of deviations from the model curves (at energies of about double the peak energy) wrong spectral fit results are expected. To produce pile-up free spectra the central part of the PSF for bright point sources should be excluded, e.g. select a ring instead of a circle via **evselect**. **epatplot** can then be used to verify the event pattern fractions in the remaining area used for the spectrum. In addition to the pattern fraction plot, task **epatplot** also calculates two diagnostic numbers which may be used to assess the presence of pile-up: In the absence of pile-up, the 0.5 - 2.0 keV observed-to-model singles and doubles pattern fractions ratios should both be consistent with 1.0 within statistical errors (1  $\sigma$  errors are given). If pile-up is present, the singles ratio will be smaller than 1.0 and the doubles ratio will be larger than 1.0. The two numbers are printed both to the console and on the plot and are appended to the input event set as attributes `SNGL_OTM` and `DBLE_OTM` (1  $\sigma$  errors: `ESGL_OTM` and `EDBL_OTM`). Note, that low energy background noise may also result in deviations from the pattern fraction model curves in the **epatplot** plots at energies below several hundred eV. This effect can be avoided if background subtracted pattern fractions are determined (parameter `withbackgroundset=Y`).

There is a slight indication that at very large offaxis-angles (well, close to the readout-nodes = CAMEX) singles are reduced and doubles increased as the noise is much higher there. If the target source is at the nominal position – close to the center of the FOV – this does not matter, for serendipitous (weak) off-axis sources the intensity may be too low anyway for significant effects on the pattern distribution. Differences may be visible if you have a bright source close to the CAMEX. For EPIC pn the task takes into account the spatial dependence of the pattern fractions.

The default behaviour is to select only events with `FLAG=0`, this can be changed via setting `withflag=N` (default: `Y`). Invalid EPIC pn patterns are also shown to indicate how much energy is lost by pattern pile-up.

If one is interested in the spatial distribution of events with intrinsic `FLAG=0` then set `withoutoutputmask=Y`: This reads the bad pixel extension(s) and the offset column extension(s) and creates a CCD map where all regions are set to 0 that have an intrinsic value of `FLAG>0`, i.e. events on bad (hot, dead, etc.) pixels, next to bad pixels, out of the FOV, close to the read-out window, on and next to offset columns. If the task finds an extension `BADPIX` then it assumes that this is an intermediate single chip event file as created during the pn chain processing, it tries to determine the relevant CCD number from the FITS header to set the FOV accordingly. If it finds extensions of the form `BADPIXnn` it takes `nn` as the CCD number. If no bad pixel extension is found at all the setting `withoutoutputmask=Y` is completely ignored as then the task cannot reliably determine the bad pixel contents of the file. Similarly for intermediate single chip files `OFFSETCO` and `OFFSETS` are accepted, for final event lists and products derived from those `OFFSETnn` and `OFFSETS` are read (the latter is then searched for the CCD number column). If no offset extension is found these flags are not used for the output map as this information cannot be reliably retrieved from the events file in general. An event file (final or intermediate) produced with **epchain** should have all the relevant information included (if not explicitly switched off, of course). The parameter `outmaskname` is parsed for a CCD number placeholder (`##`), if not found then the corresponding CCD number is appended to this parameter value (it cannot be predicted a priori how many output maps have to be created).

## 4 Parameters

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

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

<sup>2</sup>For EPIC pn single events dominate the spectrum due to the relatively large pixel size of  $150\mu\text{m}$ . These have narrower line widths than doubles. Therefore most observers will choose singles (`PATTERN==0`). In the case of pattern pile-up you lose singles and gain doubles (and later triples etc).

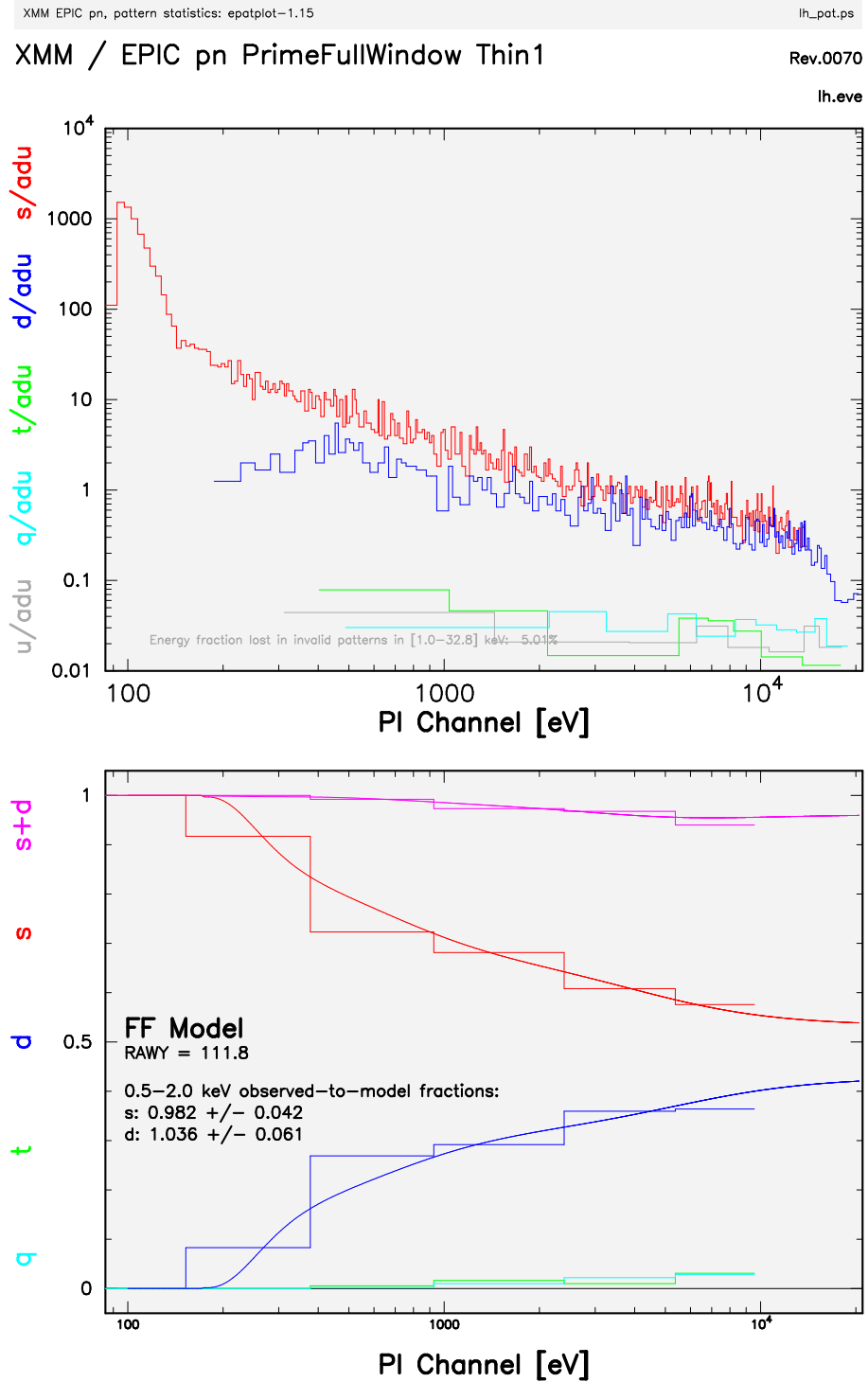


Figure 1: Example of **epatplot** output using a weak source from the Lockman Hole (Rev. 70) observation

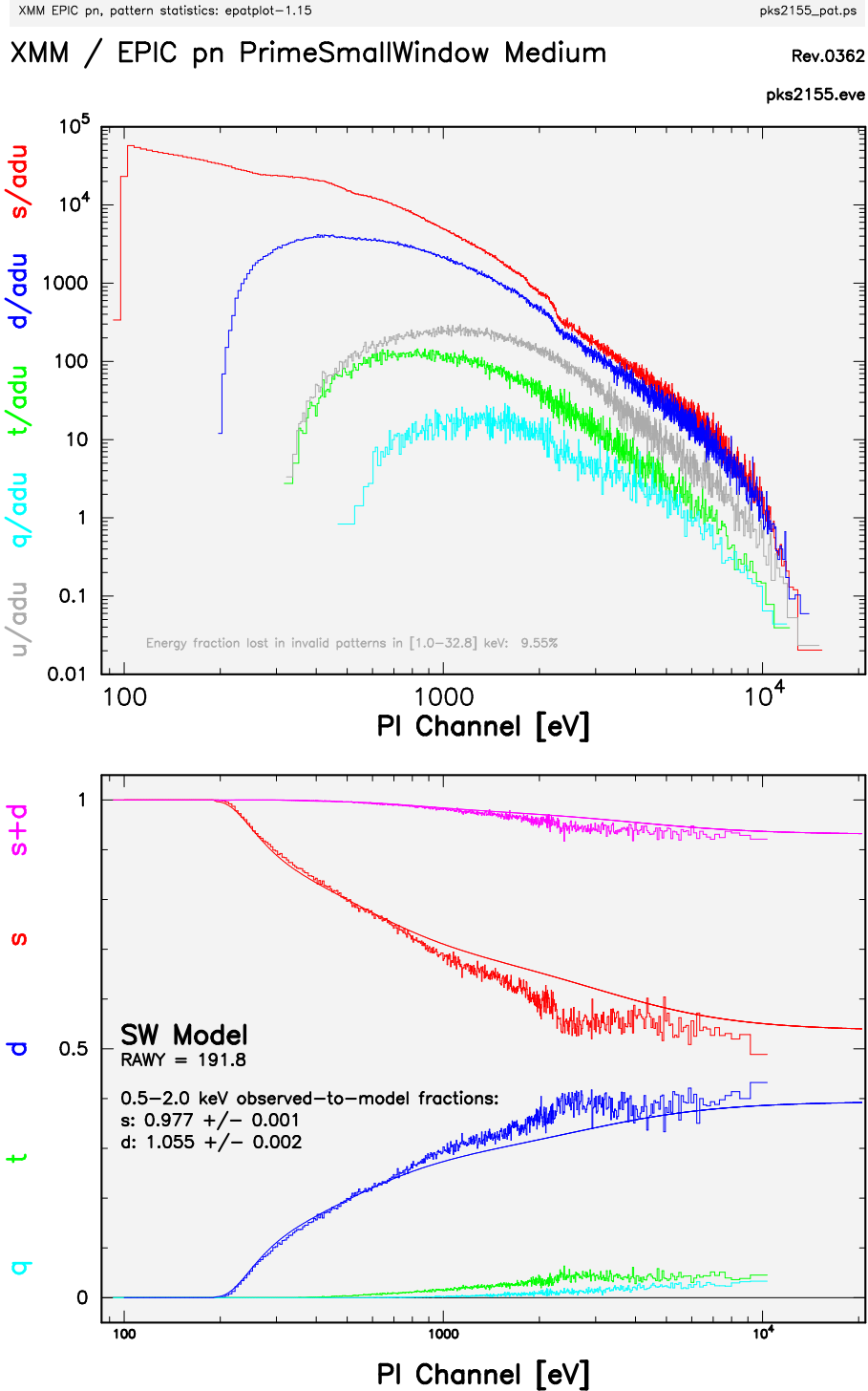


Figure 2: Example of `epatplot` output using a bright source (PKS2155). Pile-up is obvious in the 0.7 – 4 keV range (note that the observed-to-model fractions significantly deviate from 1.0).

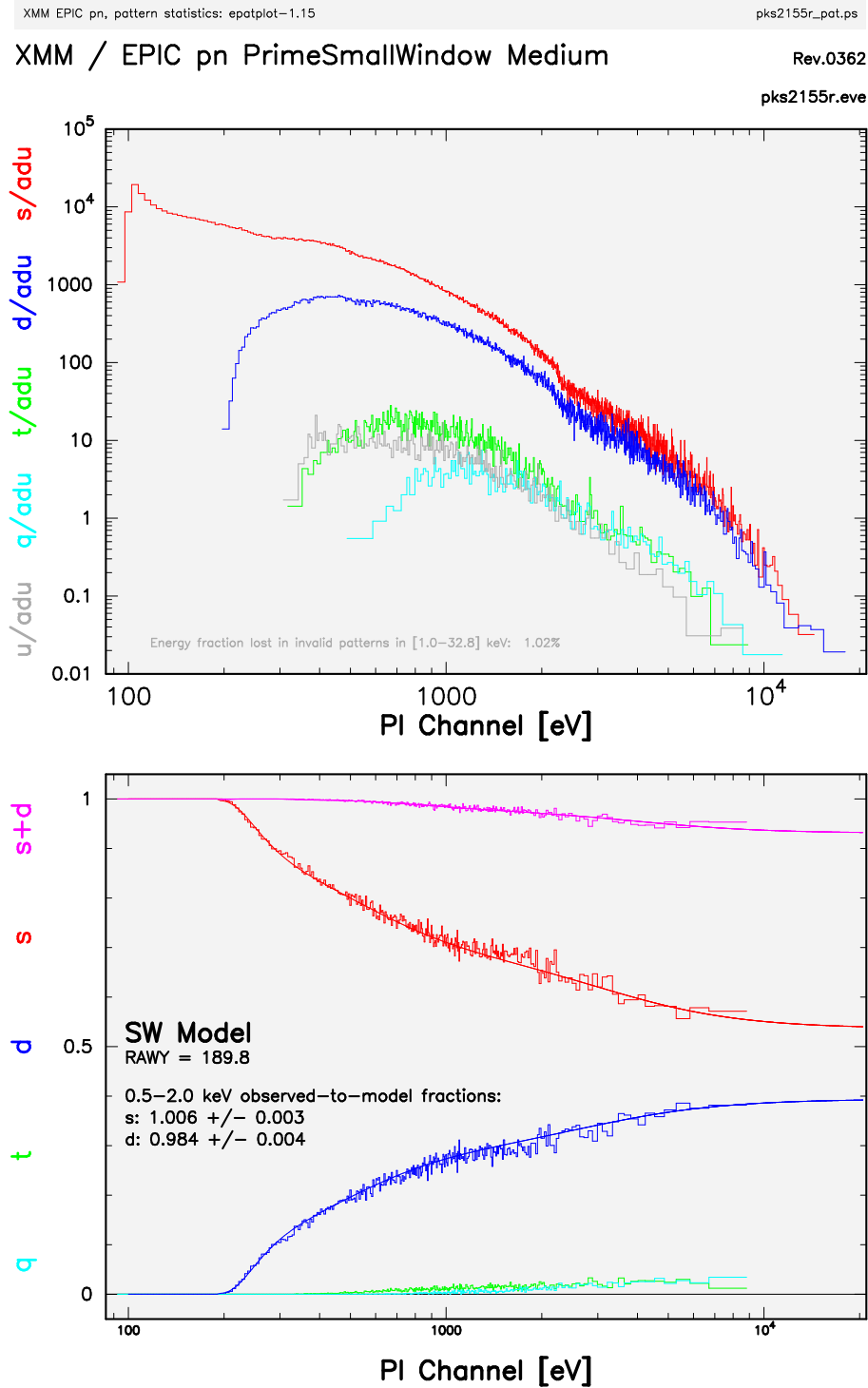


Figure 3: Example of **epatplot** output using the same data as in the previous plot but excluding the central 20 arcsec radius of the source, the pile-up is gone (the observed-to-model fractions are now consistent with 1.0).



<b>set</b>	yes	data-set	events.fits	
------------	-----	----------	-------------	--

Name of input EPIC pn or EPIC MOS events file (e.g., of a source or selected region)

<b>withbackgroundset</b>	no	boolean	N	Y/N
--------------------------	----	---------	---	-----

whether to use background event set for background subtraction

<b>backgroundset</b>	no	data-set	bkg_events.fits	
----------------------	----	----------	-----------------	--

Name of background EPIC pn or MOS event file

<b>backscal</b>	no	real	1.0	>0.0
-----------------	----	------	-----	------

background-to-source extraction area ratio; also needs to include exposure ratio if source and background exposures differ

<b>withflag</b>	no	boolean	Y	Y/N
-----------------	----	---------	---	-----

whether to select FLAG==0 internally

<b>withoutputmask</b>	no	boolean	N	Y/N
-----------------------	----	---------	---	-----

whether to create a CCD map with positions FLAG==0 set to 1

<b>outmaskname</b>	no	string	./flag0_map-##.dat	
--------------------	----	--------	--------------------	--

name of output FLAG=0 map if **withoutputmask**=Y, the substring ## will internally be replaced with the CCD number currently in use (two digits)

<b>device</b>	no	string	/VCPS	/VCPS
---------------	----	--------	-------	-------

PGPLOT output device (other options: /GIF or /XW for screen output)

<b>outdir</b>	no	string	./	
---------------	----	--------	----	--

output directory for output file

<b>useplotfile</b>	no	boolean	N	N/Y
--------------------	----	---------	---	-----

whether to use **plotfile** as output file name, otherwise construct name from input data set + '\_pat.ps'

<b>plotfile</b>	no	string	events_pat.ps	
-----------------	----	--------	---------------	--

output file name



<b>withqdp</b>	no	boolean	N	N/Y
----------------	----	---------	---	-----

whether to create also QDP output file, construct name from input data set + ‘\_pat.qdp’

<b>withdetxy</b>	no	boolean	N	N/Y
------------------	----	---------	---	-----

whether to plot also average DETX and DETY positions

<b>withsrcxy</b>	no	boolean	Y	N/Y
------------------	----	---------	---	-----

whether to plot also real-valued SRCPOSX and SRCPOSY positions

<b>withusermode</b>	no	boolean	N	N/Y
---------------------	----	---------	---	-----

whether to use the user-defined mode **usermode** instead of the SUBMODE model for comparison

<b>usermode</b>	no	integer	0	0-5
-----------------	----	---------	---	-----

if **withusermode**=Y then use this mode for model comparison, in **cal** convention:  
0=FF, 1=eFF, 2=LW, 3=SW, 4=TI, 5=BU

<b>withuserrawy</b>	no	boolean	N	N/Y
---------------------	----	---------	---	-----

whether to use the user-defined mode **usermode** instead of the SUBMODE model for comparison

<b>userrawy</b>	no	real	190.0	1-200
-----------------	----	------	-------	-------

if **withuserrawy**=Y then use this coordinate value for the spatial dependence

<b>xaxisadu</b>	no	boolean	N	Y/N
-----------------	----	---------	---	-----

whether to use ADU (instead of eV) as unit for x-axis

<b>sigma</b>	no	real	3.0	0-
--------------	----	------	-----	----

sigma to accumulate for 1 bin

<b>ccdlimits</b>	no	integer-list	1 64 1 200 1 12	
------------------	----	--------------	-----------------	--

to select certain RAWX RAWY CCDNR range from the PN (only) event file for accumulation of the distributions



<b>plotyrange</b>	no	real-list	0 0	
-------------------	----	-----------	-----	--

to fix the Y range (in log10 values) in the distribution plot for an easier comparison with other observations; if second value is not larger than first value then do an automatic determination of the Y range

<b>pileupnumberenergyrange</b>	no	real-list	500 2000	
--------------------------------	----	-----------	----------	--

energy range [eV] in which the pile-up diagnostic number is calculated

<b>modifyInSet</b>	no	boolean	Y	Y/N
--------------------	----	---------	---	-----

if N: skip writing of header keywords to increase speed

## 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.

### **noEvents** (*error*)

No EVENTS extension in input data set

### **noPattern** (*error*)

No PATTERN column in input data set

### **noPi** (*error*)

No PI column in input data set

### **emptyInput** (*error*)

No events in input data set

### **isnotEPIC** (*error*)

Instrument is not EPIC, model curves are available only for EPIC

### **noSignal** (*error*)

No net signal in input set. Wrong backscal parameter ?

### **noSignal** (*warning*)

"No net signal in observed-to-model fractions band. Wrong energy range ?  
*corrective action:* set appropriate energy band

### **noRAWX** (*warning*)

No RAWX column in input data set  
*corrective action:* continue

### **noRAWY** (*warning*)

No RAWY column in input data set  
*corrective action:* use mean value for chip



**sigmaTooLarge** (*warning*)

not enough statistics for at least 1 bin for this pattern type  
*corrective action:* continue or decrease sigma

**noFlag** (*warning*)

No FLAG column in input data set but requested via `withflag=Y`  
*corrective action:* continue without FLAG selection

**noCCDnr** (*warning*)

No CCDNR column in input data set  
*corrective action:* continue, unusual event file

**noBADPIX** (*warning*)

No bad pixel extension found in input data set  
*corrective action:* continue without requested flag output map

**noOFFSET** (*warning*)

No offset column extension found in input data set  
*corrective action:* continue without these flags in requested output map

**isnotEPN** (*warning*)

parameter setting only implemented for EPIC pn  
*corrective action:* continue with `xaxisadu=N`

**InvalidUserMode** (*warning*)

Usermode>5 not supported for PN  
*corrective action:* set correct usermode

## 6 Input Files

1. EPIC pn or EPIC MOS events file (after any selection, e.g. source region) with at least extension EVENTS, columns PI and PATTERN, if `withflag=Y` then also column FLAG; if column RAWY is present then the average Y-position is used as well; keyword SUBMODE (if present) is used to determine the relevant model pattern distribution (for EPIC pn), keyword FILTER (if present) is only used as plot label, keyword INSTRUME (if present) is used to determine instrument specific quantities (not yet in current implementation).  
If `withoutoutputmask=Y` then also bad pixel (BADPIX, BADPIXnn) and offset column (OFFSETCO, OFFSETnn, OFFSETS) extensions are looked for.
2. optional background event set

## 7 Output Files

1. PostScript file with plot of pattern distribution and pattern fraction as function of energy and pattern type (single, double, triple, quadruple event)
2. ASCII file (QDP format) with the above information if requested via `withqdp=Y`
3. FITS images for each contained CCD with regions fulfilling FLAG=0 set to 1 if requested via `withoutoutputmask=Y`



## 8 Algorithm

```
accumulate
  all valid singles
  all valid doubles
  all valid triples
  all valid quadruples
rebin each pattern type separately to have at least SIGMA significance per bin
plot distribution as function of energy for each pattern type
rebin pattern fraction to have at least SIGMA significance per bin for singles
plot fraction as function of energy for each pattern type
overplot model pattern fraction used in the DRM
```

## 9 Comments

- The task uses the package PGPLOT.

## References