EPIC Source Detection

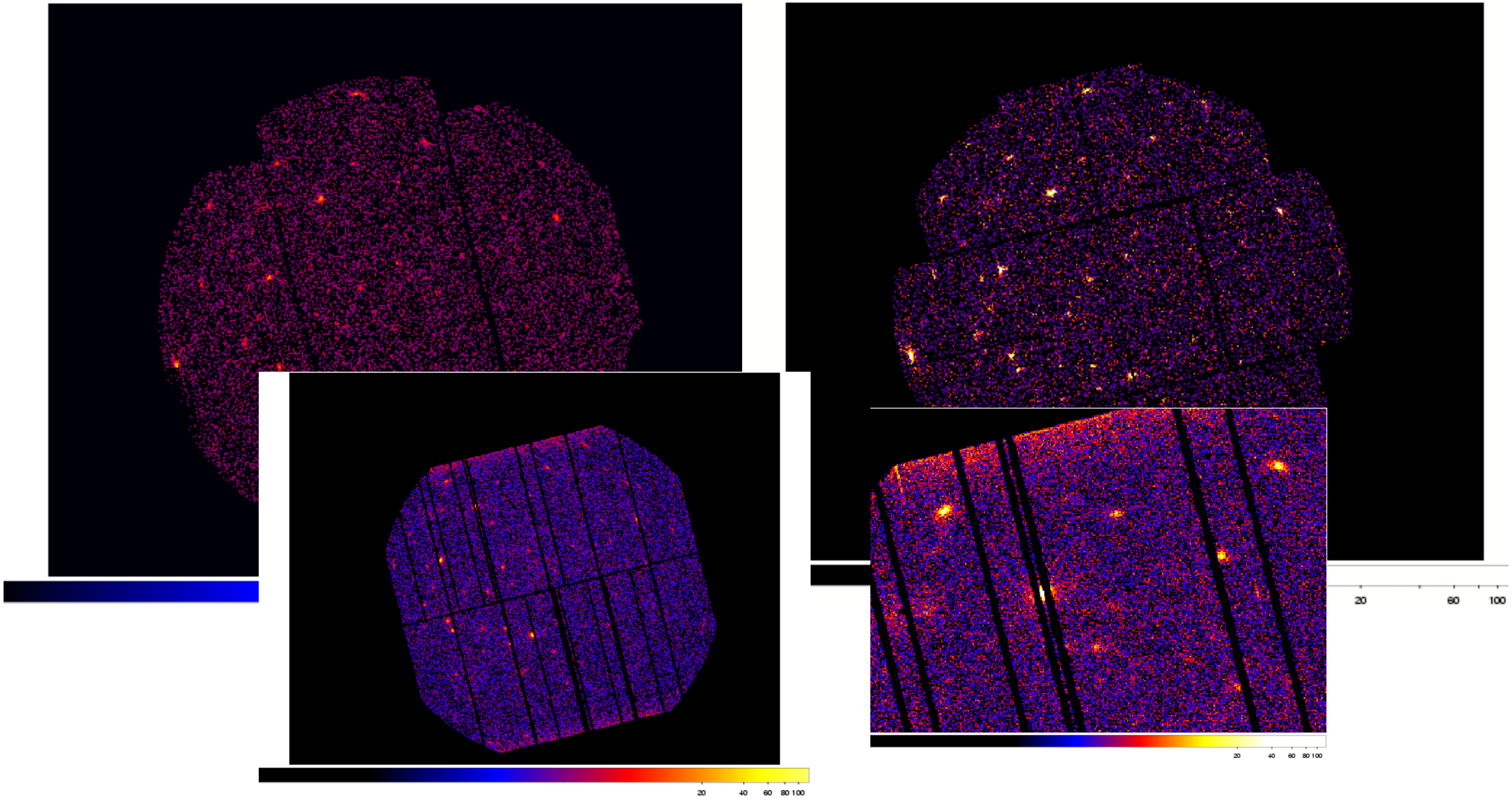
Carlos GABRIEL

XMM-Newton Science Operations Center - ESA

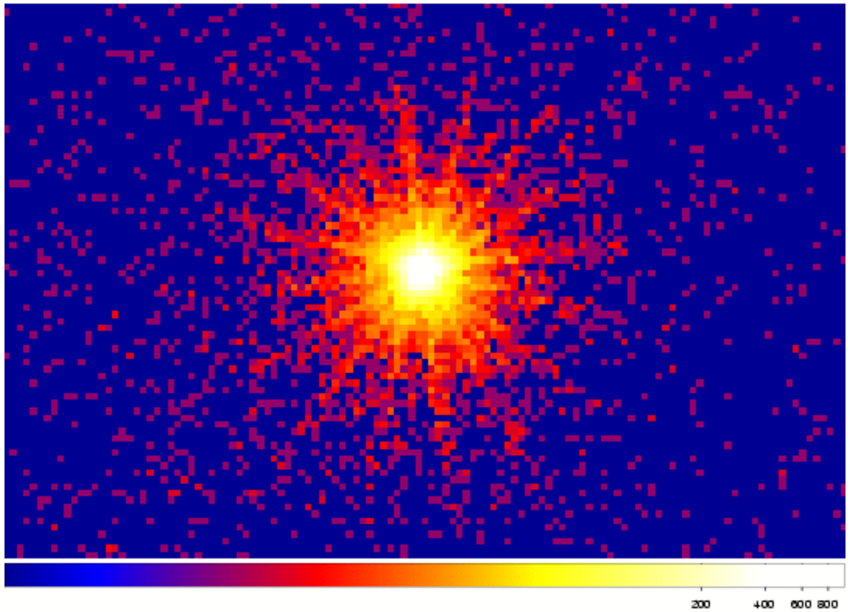
(with main contributions from G. Lamer, AIP and A. Read, UoL + SSC)

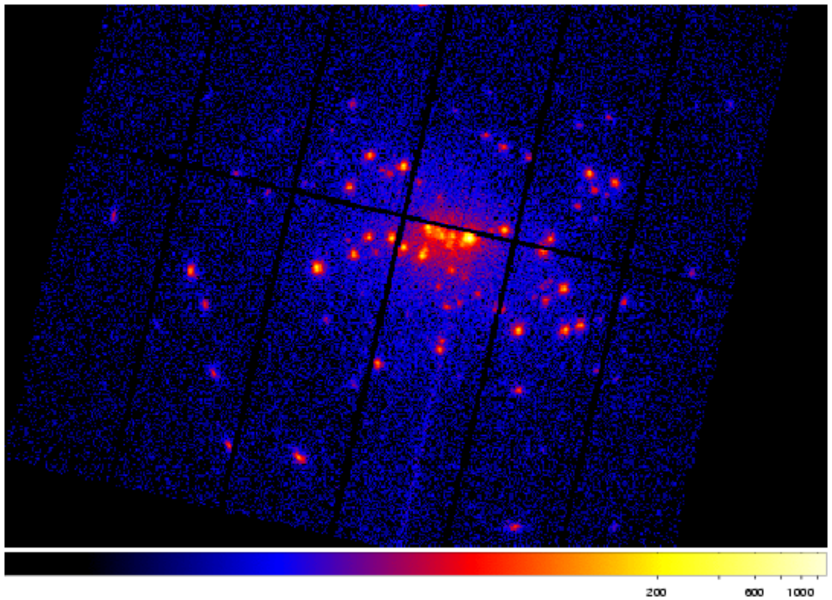
In the beginning there are images

Image production from events lists = collapsing events onto X-Y plane



In the beginning there are images





Source searching

>> source searching means basically looking for
* **significant fluctuations,** which are
* **compatible with sky sources**,
* lying on top of *more* or *less* **smooth distributions**,
* avoiding to get *fooled* by **detection defects**

>> goal:

to **maximize** source detection sensitivity
minimizing number of fake detections

Source searching: preparatory steps

Looking for (small) fluctuations on top of distributions

>> **maximization of S/N ratio** for sources to be found

... cleaning calibrated event lists against flaring periods

>> produce high energy background lightcurves

+ define a threshold + produce GTIs

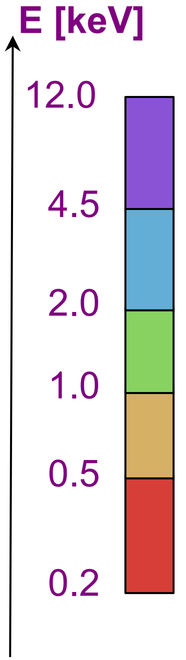
... taking into account the different source spectral characteristics

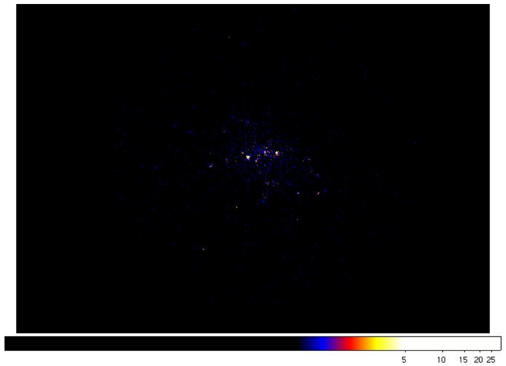
>> apply **band-passes** for deriving corresponding **images**

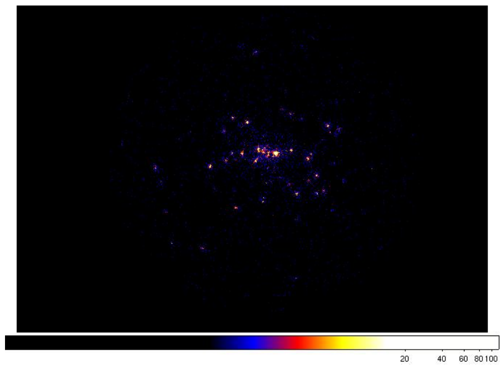
Source searching: preparatory steps

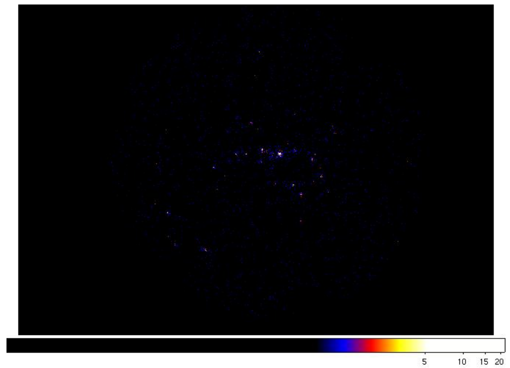
>> apply band-passes for deriving corresponding images

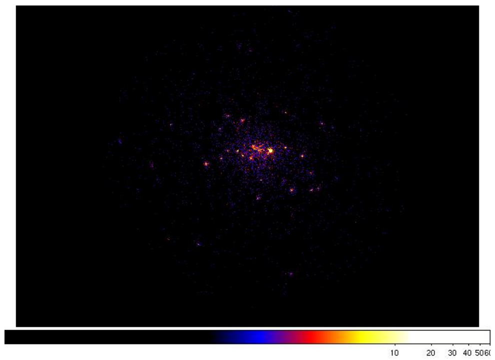
- selection of energy bands depend on the scientific purpose:
- basic XMM-Newton bands:

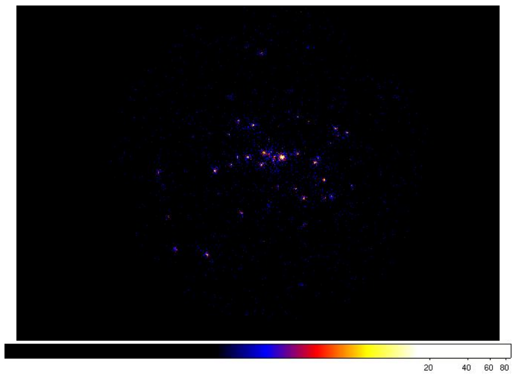












Source detection tasks

Two methods of performing source detection on EPIC datasets:

1) edetect_chain

perl script running all these tasks consecutively

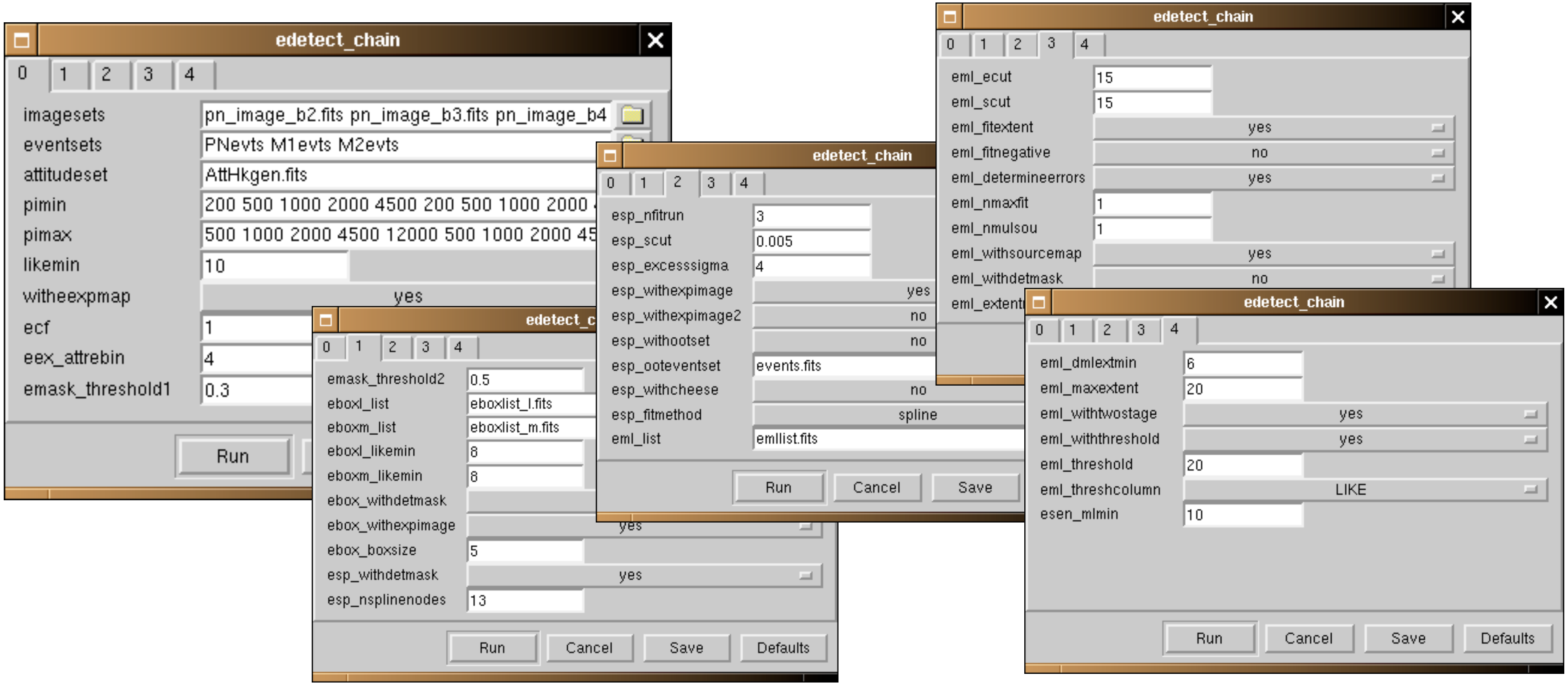
Task	Purpose	input data sets	output data sets
eexpmap	creation of exposure maps	images, attitude files	exposure maps
emask	creation of detection masks	exposure map	detection mask
eboxdetect (local mode)	sliding box detection	images, exposure maps, detection mask	local box list
esplinemap	creation of background maps	images, exposure maps, detection mask, local box list	background map
eboxdetect (map mode)	box detection using bkg map	images, exposure maps, detection mask, background maps	map detect source
emldetect	maximum likelihood fitting	images, exposure maps, detection mask, background maps	final source list
esensmap	creation of sensitivity maps	exposure map, detection mask, background map	sensitivity map

2) ewavelet mexican hat wavelet algorithm for detecting both point and **extended** sources.
Easy to use and efficient, but less reliable source parameters than those from edetect_chain

edetect_chain

edetect_chain is able to work on up to 240 images at one time

→ eg (“PPS approach”): running simultaneously the whole detection chain with 15 input images corresponding to the 5 standard energy bands of each EPIC camera



Source detection tasks 1 x 1

To quantify the significance of detected signals we need to know the observing conditions, eg., how long we have exposed the different parts of the detectors

>> effective exposure time for each detector point

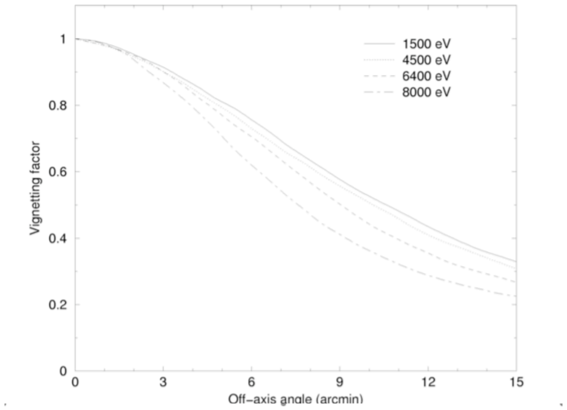
spatial quantum efficiency, filter transmission, mirror vignetting and FOV
calculating for each attitude bin the exposure values projected onto the sky

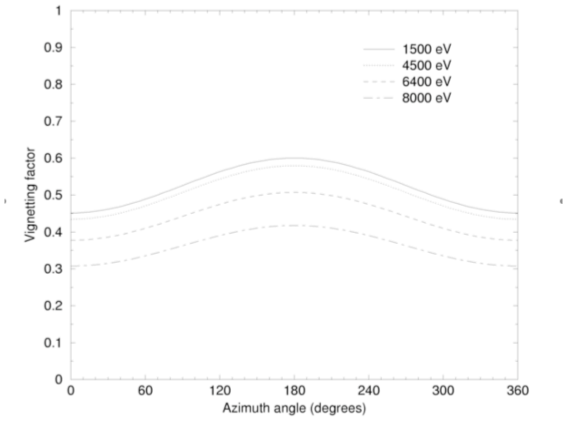
[data taken from Calibration Files]

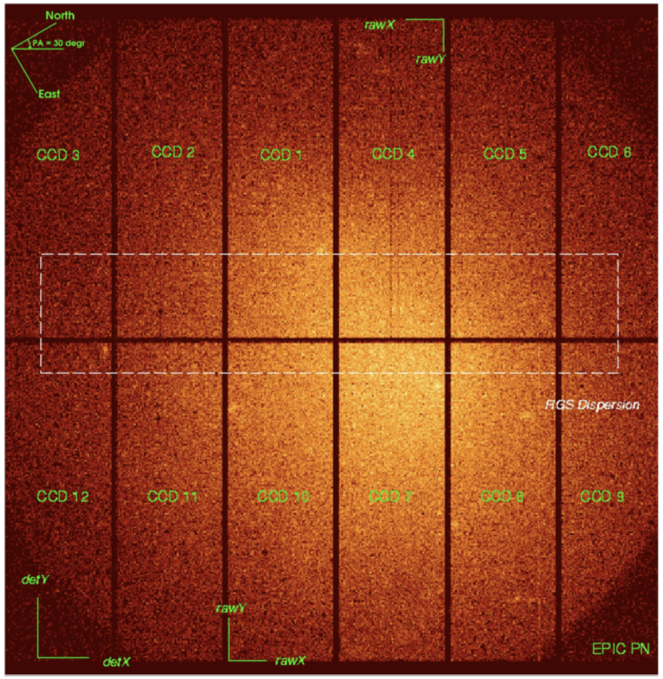
>> **instrument maps** containing **spatial efficiency** of the instrument (unit=[seconds])

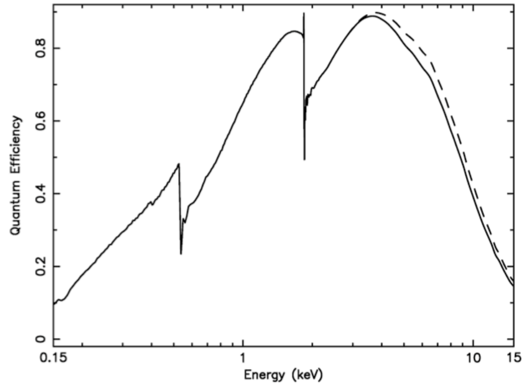
Source detection tasks

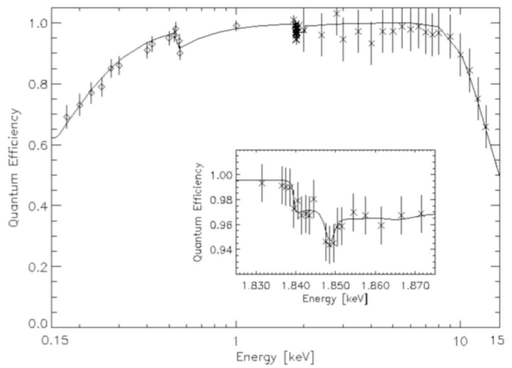
spatial quantum efficiency, filter transmission, mirror vignetting and FOV calculating for each attitude bin the exposure values projected onto the sky











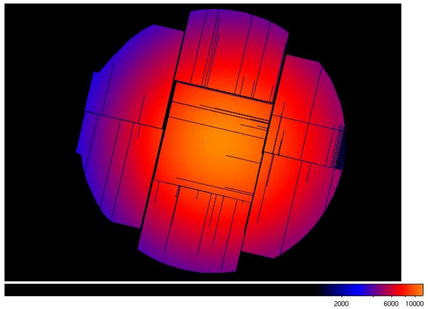
edetect_chain: eexpmap

(PN) example of single task commands issued:

1) create the multiband exposure maps by eexpmap:

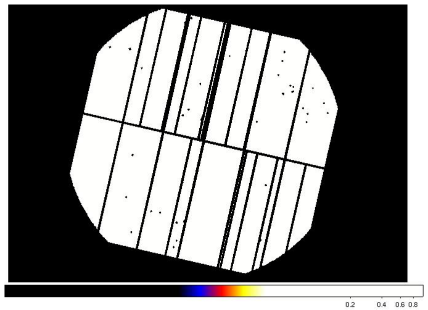
* event energy is assumed to correspond to PI channel boundaries given (parameter *pimin* and *pimax*)
* event pattern types for quantum efficiency calculation also taken into account (parameter *pattern*)
* output in detector or sky coordinates (same as input image)

eexpmap *attitudeset*=xxxATTxxx.FIT *eventset*=xxxEVLIxxx.FIT *imageset*=xxxIMAGExxx.FIT *pimin*="200 500 1000 2000 4500" *pimax*="500 1000 2000 4500 12000" *expimageset*="**pn_1exp.fits pn_2exp.fits pn_3exp.fits pn_4exp.fits pn_5exp.fits**"



2) create the detection maps
(area defined which is suitable for source detection):

emask *expimageset*=pn_2000.fits *threshold1*=0.5 *detmaskset*=**pn_mask.fits**



Main criterium - valid area is only **area_i** such that:

exposure_i > *threshold1* * MAX(**exposure**)

finding source candidates

esa

Source detection is performed by eboxdetect

3) sliding box detection (local mode):

`eboxdetect` in "local mode"

a) **source counts** accumulated in 3x3 or **5x5** pixel window (parameter boxsize)
background from surrounding 7x7 or **9x9** pixels (40 or 56 pixels respectively)

Detection of extended sources doubling up to 3 times the pixel size in consecutive runs

Background subtracted source counts calculated applying correction factors to account for respective **fractions** of **source** counts falling on **source and background area**

n = detection box size

Enboxed energy fractions in source / background box: $\boldsymbol{\alpha} = \sum_{n\times n} \mathrm{PSF}$ / $\boldsymbol{\beta} = \sum_{(n+4)\times(n+4)} \mathrm{PSF} - \sum_{n\times n} \mathrm{PSF}$

Raw box counts & raw background counts: $\mathbf{C} = \sum_{n\times n} \mathrm{image}$ & $\mathbf{Bg} = \left(\sum_{(n+4)\times(n+4)} \mathrm{image} - \sum_{n\times n} \mathrm{image}\right) / ((n+4)^2 - n^2)$

PSF corrected and background subtracted counts: $\mathbf{SC} = \mathbf{C} - \mathbf{Bg} * n^2 \;/\; [\alpha - \beta * n^2 / ((n+4)^2 - n^2)]$

sliding box detection / I - cont.

Detection likelihoods given as:

L = -ln p $\quad$ p = probability of Poissonian random fluctuation of background counts in cell resulting in ≥ observed source counts

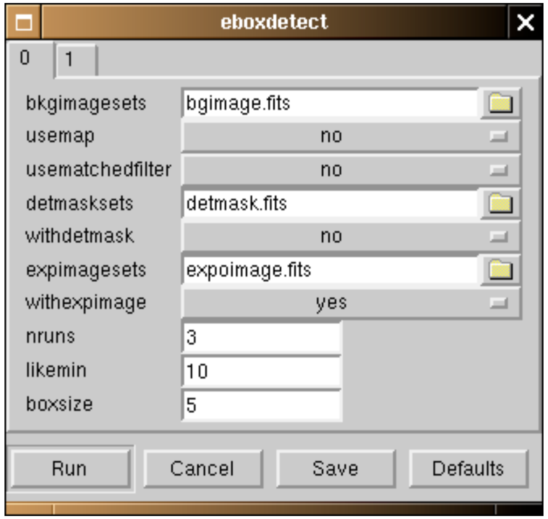
(p using incomplete Gamma function $\Gamma(a,x)$ as function of raw source and raw background counts)

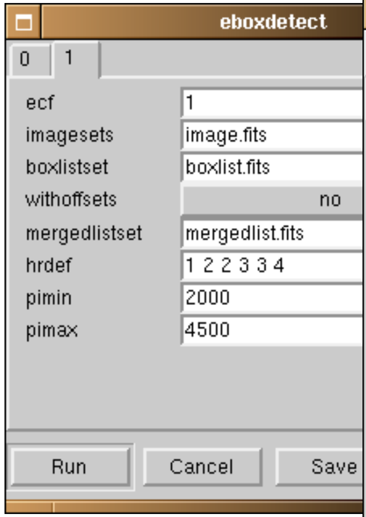
In case of simultaneous detection over several bands, likelihoods are added (!) and transformed into equivalent single band detection likelihoods

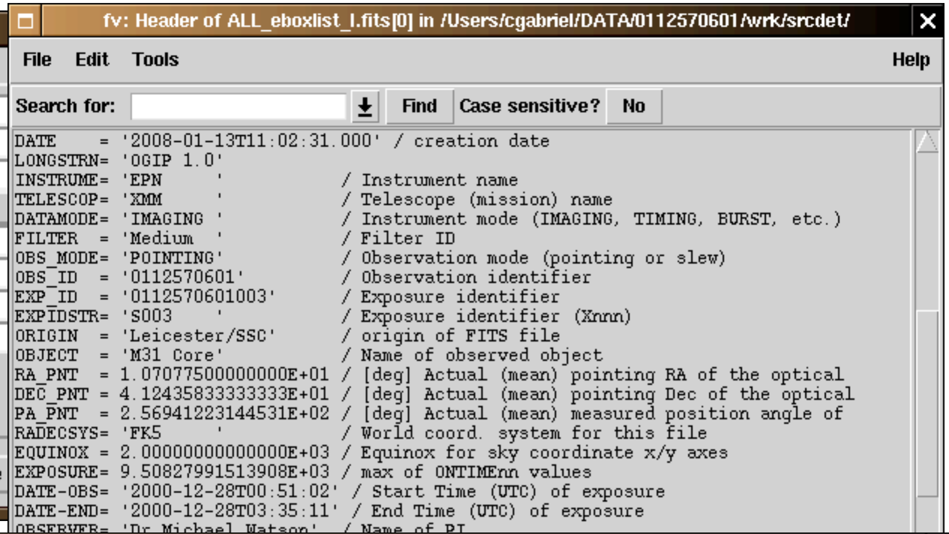
$$L = \Gamma(n_{band}, \sum_{i=1,n} L_i)$$

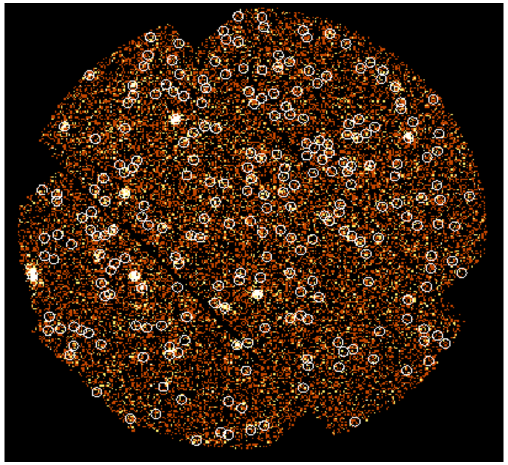
eboxdetect ***usemap*=no** *likemin*=10 *withdetmask*=yes *detmasksets*=pn_mask.fits \
imagesets="PNIM_1000.FIT PNIM_2000.FIT PNIM_3000.FIT PNIM_4000.FIT PNIM_5000.FIT" \
expimagesets="pn_1000.fits pn_2000.fits pn_3000.fits pn_4000.fits pn_5000.fits" \
pimin="200 500 1000 2000 4500" *pimax*="500 1000 2000 4500 12000" \
boxlistset=**eboxlist_ local.fits**

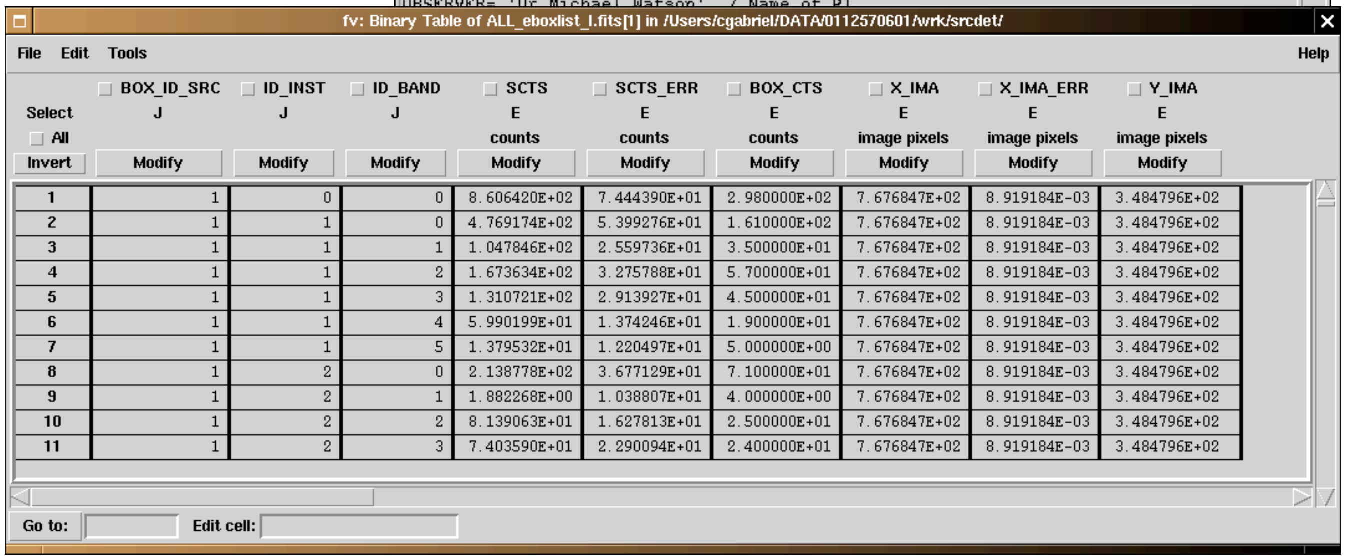
eboxdetect > FITS tables











creating background maps

esa

Modelling the background is a **key issue** for source detection

EPIC background has three main components

a) Photons:
- astrophysical, dominated by thermal emission at lower energies (unresolved cosmological sources)
- solar wind charge exchange
- single reflections from out of FOV, out-of-time events, etc

b) Particles:
- soft proton flares
- internal (cosmic-ray induced) background, direct (on CCDs) / indirect (fluorescence of S/C)

c) electronic noise
- bright pixels, columns, etc, readout noise, etc.

>> complex issue - so far in source detection by default phenomenologically:
2D-spline

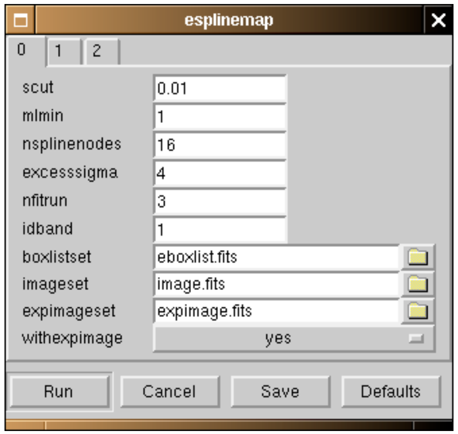
creating bkg maps: esplinemap

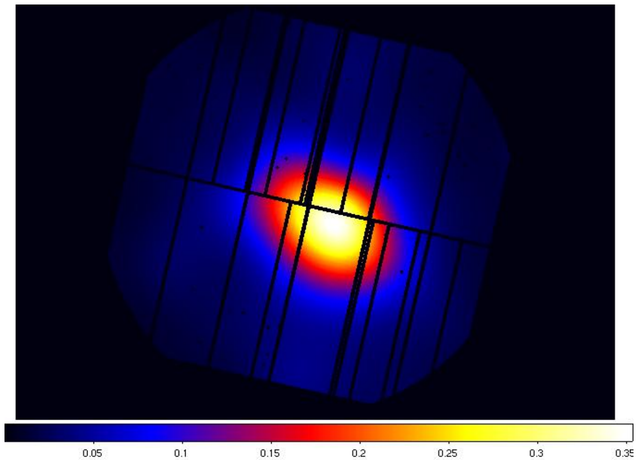
4) creation of background maps (done per detector and band)

a) Cutting-out sources (using source brightness dependent radius), `esplinemap` blanks out areas with sources detected by `eboxdetect` >> cheesed image

b) n x n (default=12) spline fits >> smoothed background map for entire image

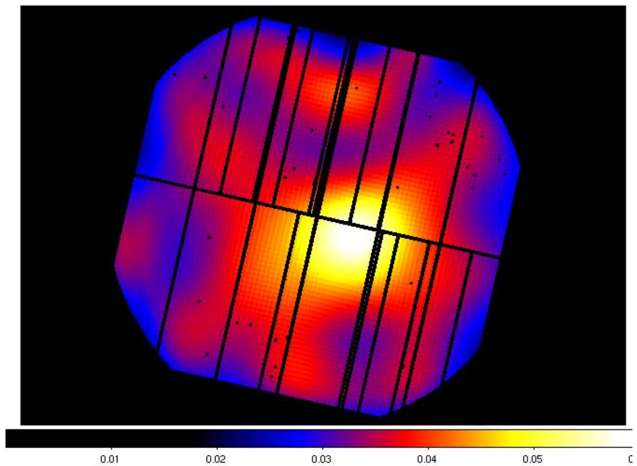
`esplinemap` *bkgimageset*=pnback_1000.fits *scut*=0.005 *imageset*=PNIM_1000.FIT *nsplinenodes*=***12*** \
withdetmask=yes *detmaskset*=pn_mask.fits *withexpimage*=yes, *expimageset*=pn_1000.fits \
boxlistset=eboxlist_ local.fits





pnback_2000.fits

pnback_5000.fits



eboxdetect in map mode

5) box detection using the background maps

in **map mode** background is taken from background maps determined by **`esplinemap`**
>> improved detection sensitivity compared to local detection map

`eboxdetect` *usemap*=yes *likemin*=8 *withdetmask*=yes *detmasksets*=pn_mask.fits \
imagesets=… *expimagesets*=… *pimin*=… *pimax*=... *bkgimagesets*="pnback_1000.fits …" \
boxlistset=**eboxlist_ map.fit**

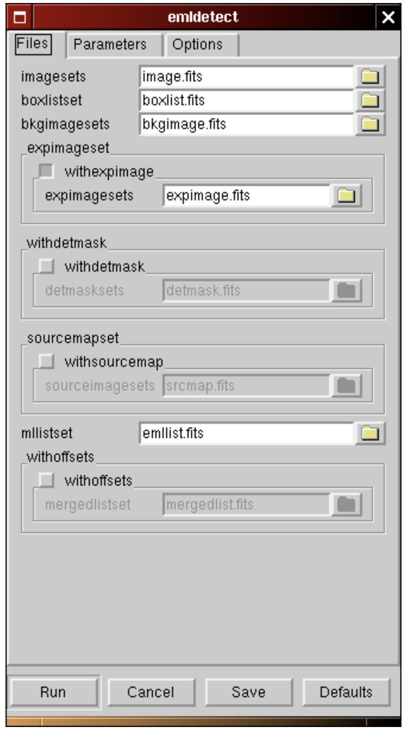
Output table:
- one raw per input image for each detected source (source table)
- number of summary rows containing broad band results for each EPIC telescope
- combined results for all EPIC telescopes taken together

Source table:
- count rates and source positions including statistical errors + fluxes + ...
- detection likelihoods (per band and total) given for each source
  $l = -\ln p$ with p=probability of random fluctuation of counts resulting in $N_{Counts} \geq C_{obs}$
- if several images inputted then hardness ratios are calculated:
  $HR_i = (B_m - B_n) / (B_m + B_n)$

max likelihood fit: emldetect

6) maximum likelihood fitting for getting final source list

emldetect *imagesets*=… *expimagesets*=… *bkgimagesets*=… pimin=… pimax=… \
boxlistset=eboxlist_map.fit *ecf*="10.596 6.8157 2.0542 0.99483 0.25933" \
mllistset=**emllist.fits** *mlmin*=10 *determineerrors*=yes



Method: **Simultaneous maximum likelihood PSF fit** to source count distribution (convolved with a **source extent model**) in all bands with following free parameters:

- source location (α, δ) | constrained to same best-fit value
- source extent (gaussian sigma) | in all energy bands of each EPIC
- source count rates in each band | individual best-fit value in each band

Second loop for fitting two PSFs to "extended sources" - if better > recalculation

PSF fitting may be performed in **single source** or in **multi-source** mode.
In multi-source mode sources with overlapping PSFs are fitted simultaneously (up to 6)

Energy conversion factors (ECF) supplied for conversion of count rates into correct flux values. **The ECFs depend on filter and pattern selection**

`emldetect` & extended sources

PSF fitting by `emldetect` is most crucial step for characterization of extended sources

Default since SAS 7.1. (and for 2XMM) is the convolution of PSF with a β model for source extent

$$f(x,y) = \left(1 + \frac{(x-x_0)^2 + (y-y_0)^2}{r_c^2}\right)^{-3\beta+1/2}$$

$\beta = 2/3$ (canonical for surface distribution of clusters)

Fitting procedure minimizes the C - statistic

$C = 2 \sum (e_i - n_i \ln e_i)$ e= expected model & n= number of counts in pixel i

Detection likelihood for each input image IM is $L_{IM} = C_{NULL} - C_{BEST}$ C_{NULL} is C of Null-hyp.

Detection likelihood obeys $L = -\ln(P)$ with P probability that source is spurious

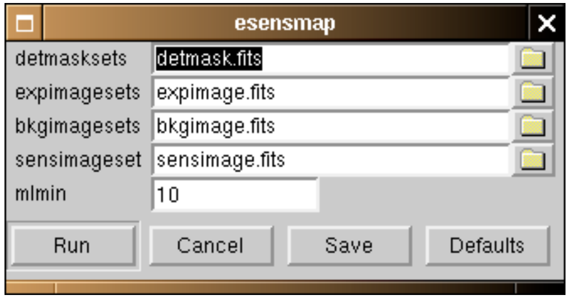
Extended likelihood L_{ext} calculated in analogy with C_{NULL} = best fitting point source model

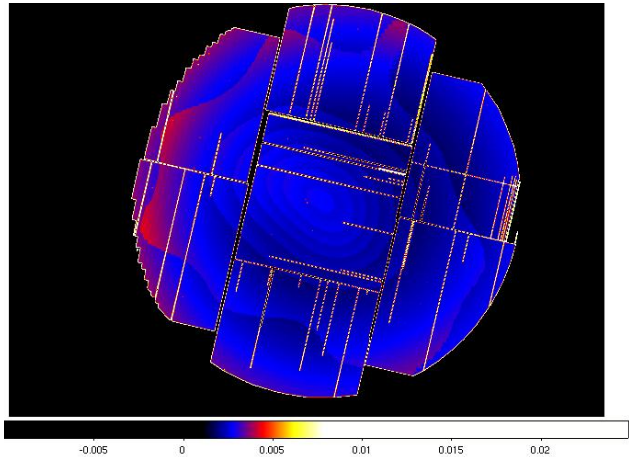
Second fitting loop against source confusion: 2 source models simultaneously fitted (only for brighter sources)

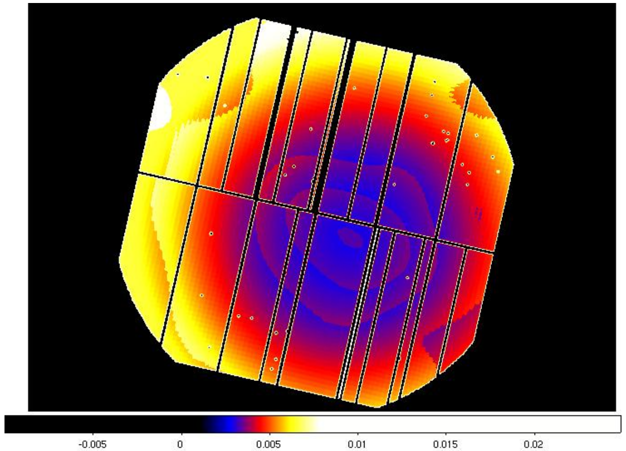
7) creation of sensitivity maps (called for each detector and band)

esensmap *expimagesets*= pn_1000.fit *bkgimagesets*=pnback_1000.fits *detmasksets*=pn_mask.fits \
mlmin=10. *sensimageset*=sens_map1.fits

Sensitivity map == point source **detection upper limits (**vignetting corrected source count rate corresponding to the likelihood of detection as specified in the parameter file) for each image pixel.







position rectification: `eposcorr / catcorr`

off-edetect_chain) ***position rectification using optical catalogues***

correlation with optical source catalogue, checking whether there are offsets in RA and DEC which optimize the correlation

so far used in SAS / PPS:

`eposcorr` *xrayset*=emllist.fits *opticalset*=usnob1.fits *findrotation*=yes *maxoffset*=10 *maxdist*=15.

new since SAS 12 / used in 3XMM-PPS:

`catcorr` *srclistset*=emllist.fits *catset*=catextract.ds *mingood*=10 *minfit*=5 *maxoffset*=10
using not only USNO-B, but also 2MASS and SDSS >> covering 85% of all observations

new in SAS 13:

`poscorr3xmm` *srclistset*=srclist.ds **- corrects an off-axis dependent systematic offset in astrometry**

rectification evaluation: `evalcorr`

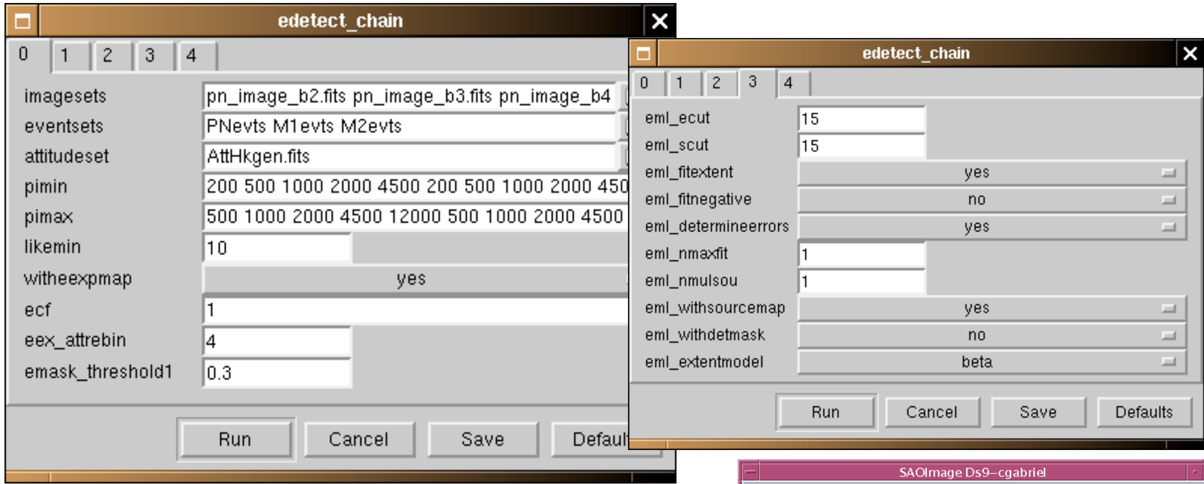
evaluates the quality of the position rectification, eg. (2XMM)

POSCOROK = True if $L > 9.0 + (2.0 * L_{\mathrm{NULL}})$

L_{NULL} = likelihood for purely coincidental X-ray / opt. matches in given obs.

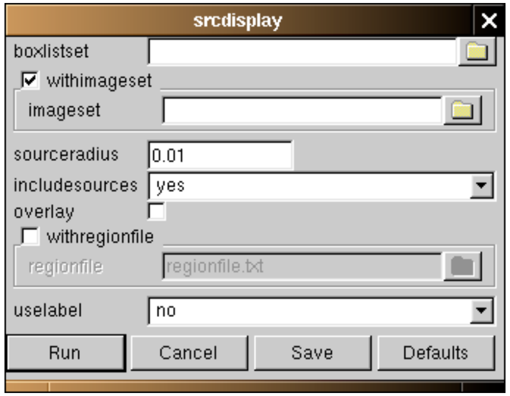
displaying sources: `srcdisplay`

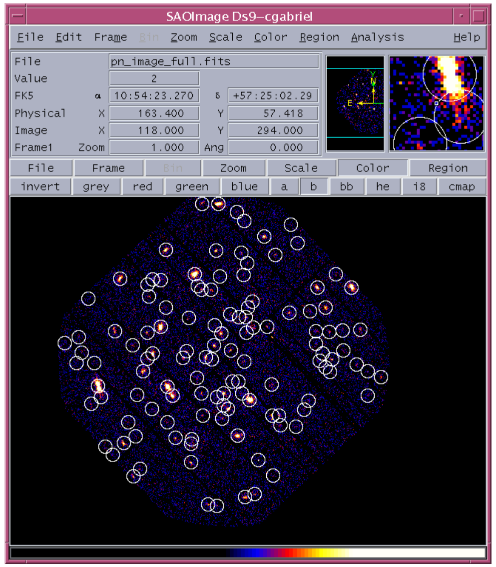
`edetect_chain –d`



source list

`srcdisplay -d`

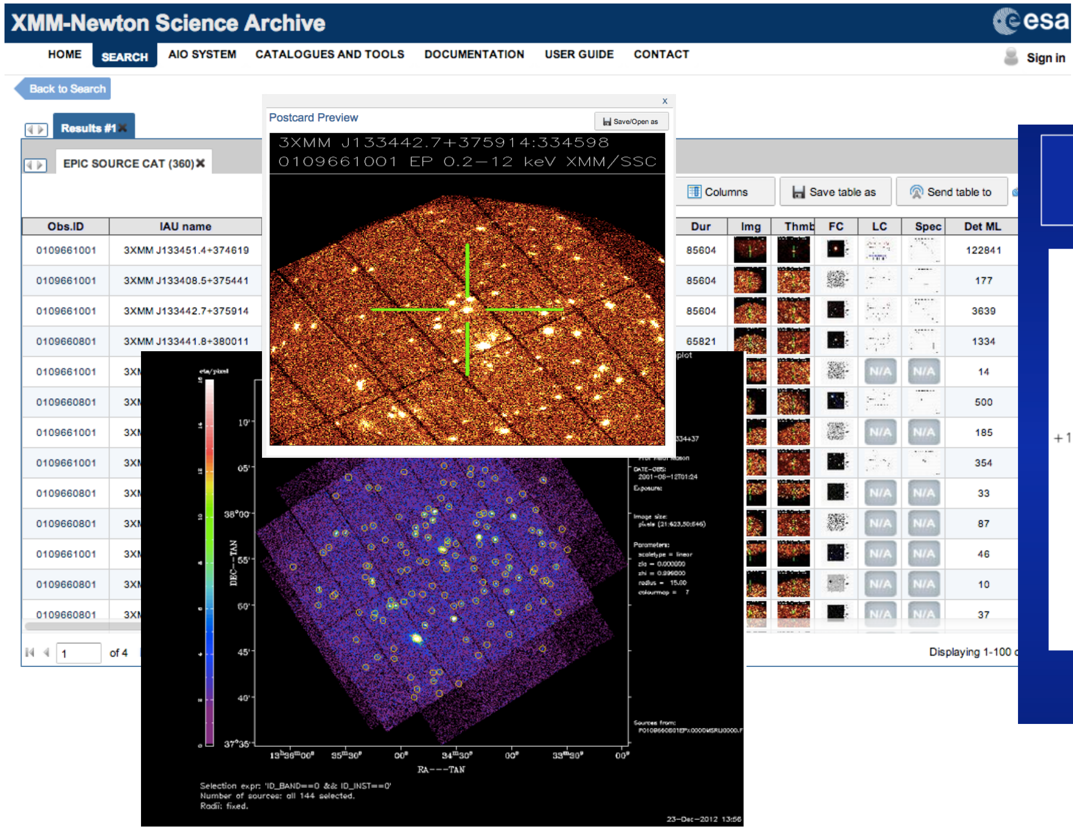


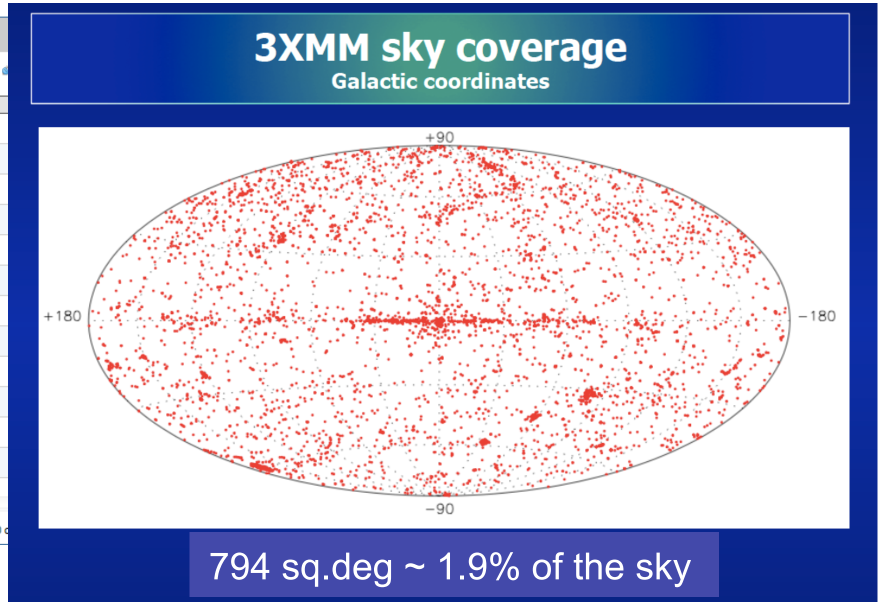


Full reprocessing + 3XMM

* Survey Science Centre (SSC) @ UoLeicester finished in 2012 the reprocessing of all the XMM-Newton data on behalf of ESA

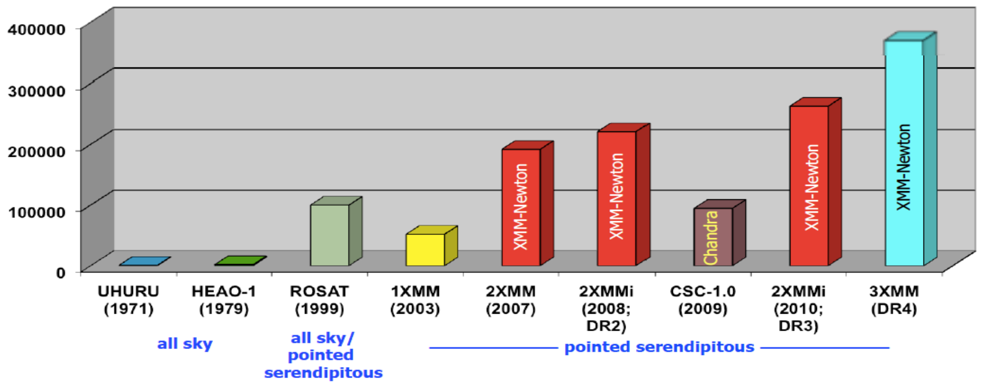
>> uniform archive in terms of processing and calibration
>> 3XMM catalogue = largest catalogue of X-ray sources = 530k detections >> ~ 370k unique sources





3XMM properties

- 7497 XMM-Newton EPIC observations used with net exposures in the range [1-130] ksec
- total sky area = 1200 deg^2 >> ~ 800 deg^2 corrected for field overlaps
- Median of counts / detection: ~ 50 counts / PN and ~ 30 counts / M1/M2
- 35 % of all detections > 100 PN counts >> sufficient for basic spectral analyses (25% M1/M2)
- X-ray flux in the range [10^{-16} - 10^{-9}] erg/cm^2/s
- Total band ([0.2-12]keV) median flux of catalogue = 2.4 x 10^{-14} (20% of fluxes below 10^{-14}) erg/cm^2/s
- average 1-sigma position error for whole catalogue ~ 1.2 arcsec



Detecting sources in overlapping fields

Main example for overlapping fields is the Mosaic Mode

Basic definitions:

- series of stable pointings with EPIC cameras in FF / Window mode, continuously taking data (same filter)
- only 1 PN offset map, taken by first pointing
- angular offset between pointings within [0.2 - 60] arcmin
- shortest integration time per pointing = 1500 sec
- whole observation included in one ODF - if observation not possible within one revolution, then several obs's.

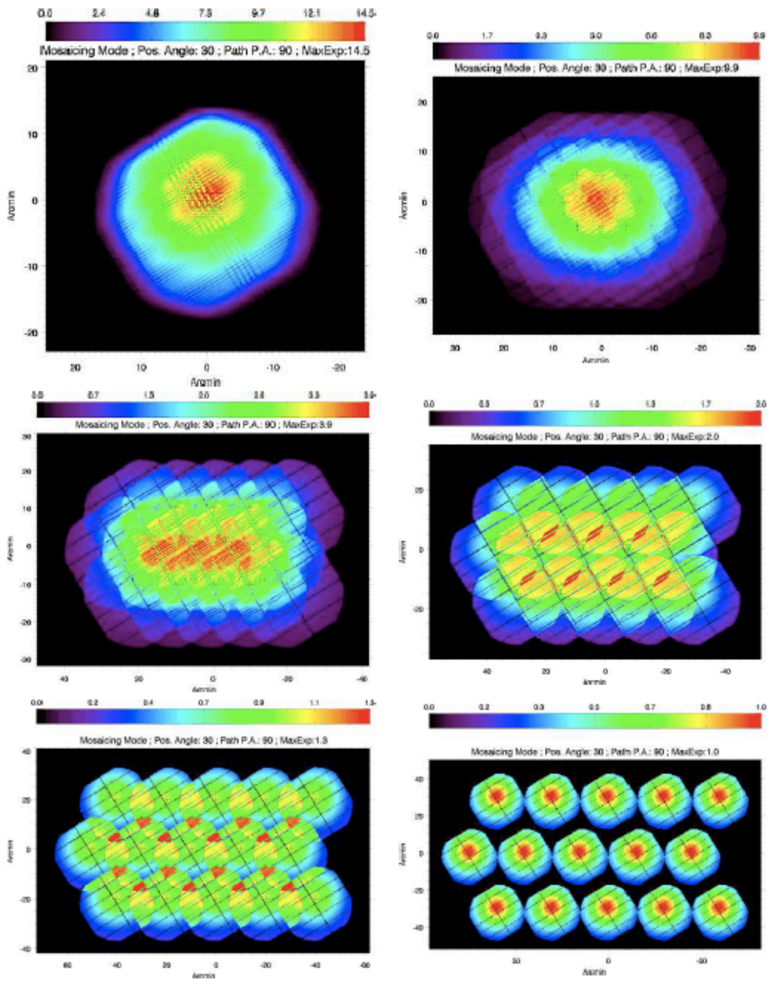


Figure 1: These figures simulate the EPIC-pn effective exposure maps achieved for a mosaic consisting in 5x3 individual pointings and for different angular offsets (1,5, 10, 15, 20 and 30 arcmin). The duration of a single pointing is taking as unit for these exposure maps. The relative position angle of the instruments has been arbitrarily set to 30degrees.

Simulated exposure maps of a 5x3 mosaic taken for angular offsets of 1.5', 10', 15', 20' and 30' with flat exposure per pointing, ignoring slews.

From P. Rodriguez TN

Mosaic mode: analysis

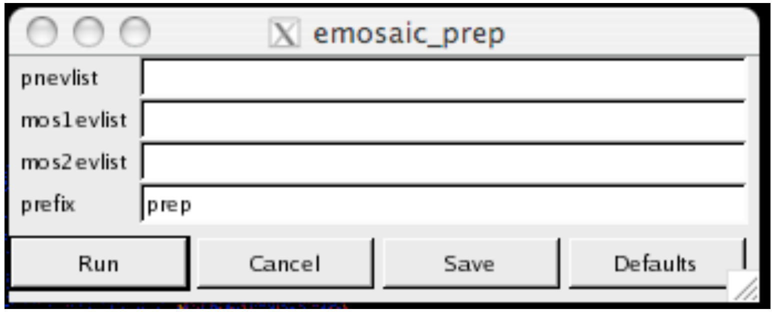
Question to SAS / PPS:
- how to treat this data?
- definitions:
  - separate data corresponding to different pointings as if they were different exposures (ignoring slews)
  - treat them coherently for source detection, eg. **one** call to `eboxdetect` (map mode) and `emldetect`

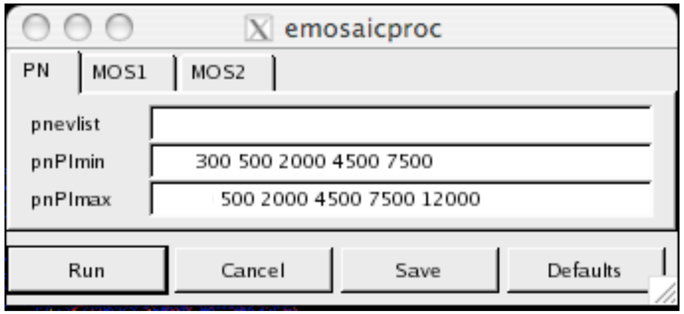
Our PPS scheme: (1) process full ODF up to the level of calibrated event list + images
(2) separate mosaic ODF into n single pointing ODFs >> process them "normally"

Our SAS scheme:
>> **normal** reduction to large **single event file** (epicproc)
>> **separation** of events from different pointings
>> one event file per point per instrument through `emosaic_prep`
>> coherent source detection of (overlapping) chosen fields through `emosaicproc`

Source detection working with all EPIC data (memory ~ map size can become an issue)

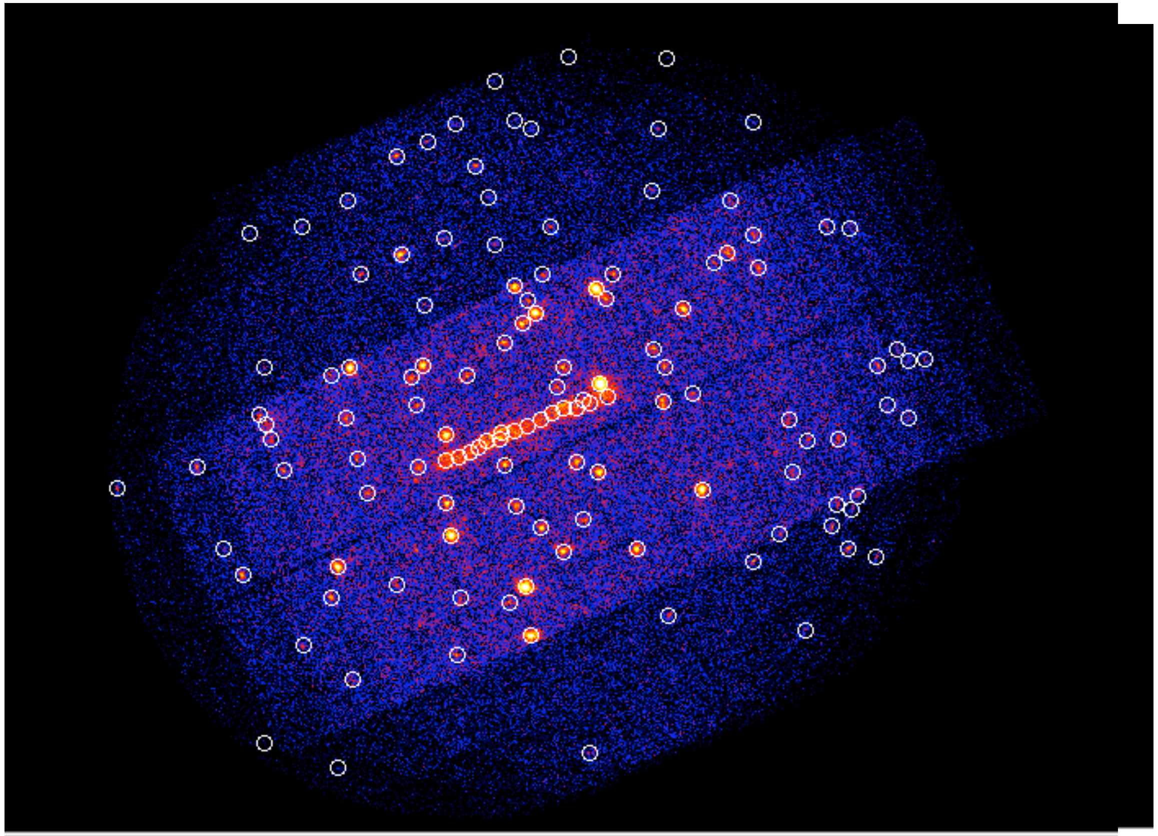
>> GUIs





Mosaic mode - results - one mosaic

Jupiter observation 0200080701 - 4 pointings:



Problems:

> number of pointings can be large
  >> **enormous** needs of memory
> combination of not overlapping data
  not necessary at all

>> decision about which points to be combined and how to do recombination of source lists left to the observer

Remember - right combination:

- more efficient source detection instead of WRONG source detection (wrong LHs if no separation)

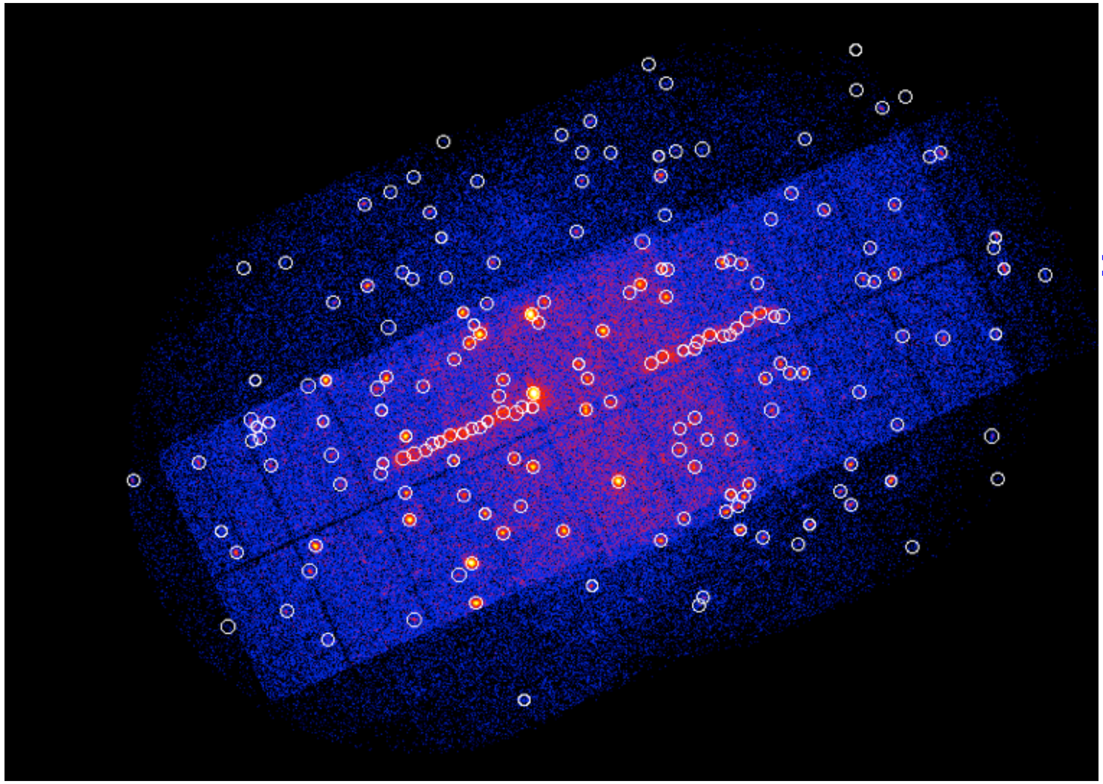
Detection using 3 spectral bands for each pointing and instrument, [400-1000], [1000-2000] and [2000-10000] eV

>> 36 images combined for eboxdetect (map mode) and emldetect

>> so far no PPS implementation

Mosaic mode - results - combining ODFs

Jupiter observation 0200080201+ 0200080701 - 4+4 pointings:



>> can be used by any overlapping observations, also with single pointing obs's taken at different times

Also using 3 spectral bands for each pointing and instrument, [400-1000], [1000-2000] and [2000-10000] eV

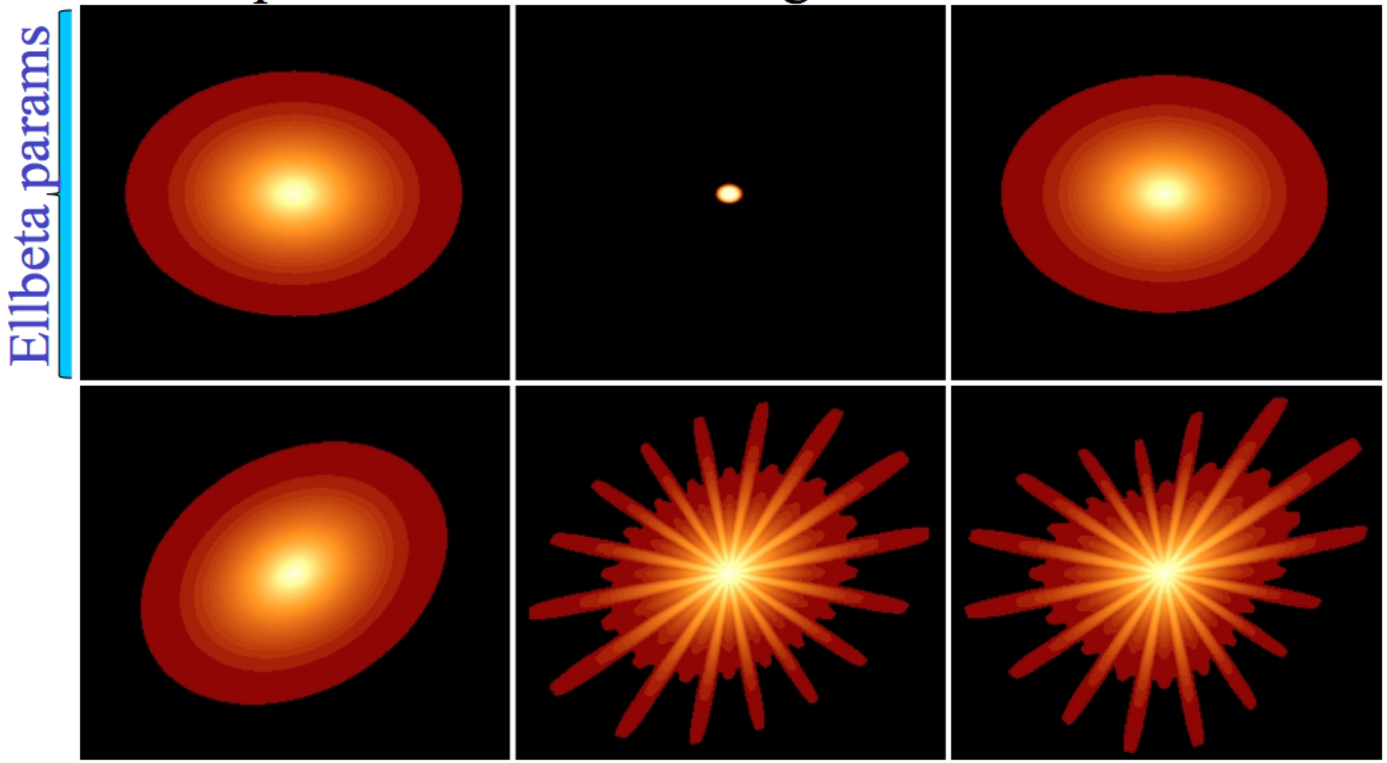
>> 72 images combined for eboxdetect (map mode) and emldetect

Default since SAS 12: 2D PSF

replaced SciSim generated PSF description through realistic model**:**

Approach already implemented as non-default in SAS10 / SAS11 is now after refinement default mode

Point Spread Function: Six stages towards a full 2-D PSF

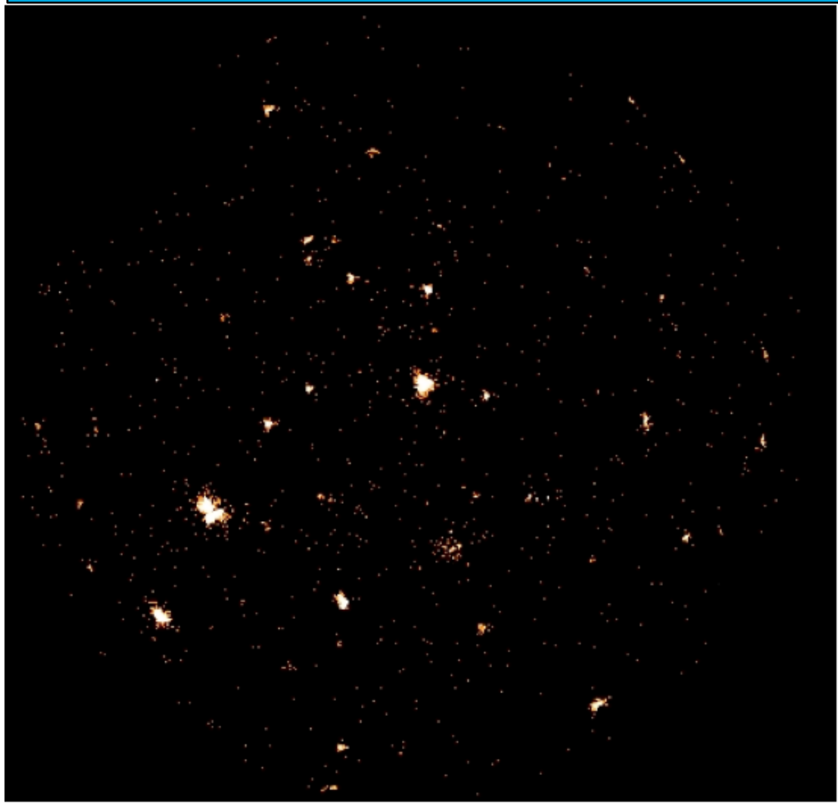


[1] Ell. PSF at given off-ax angle/energy [2] Central Gauss peak (off-ax/en) [3] Combine 1+2

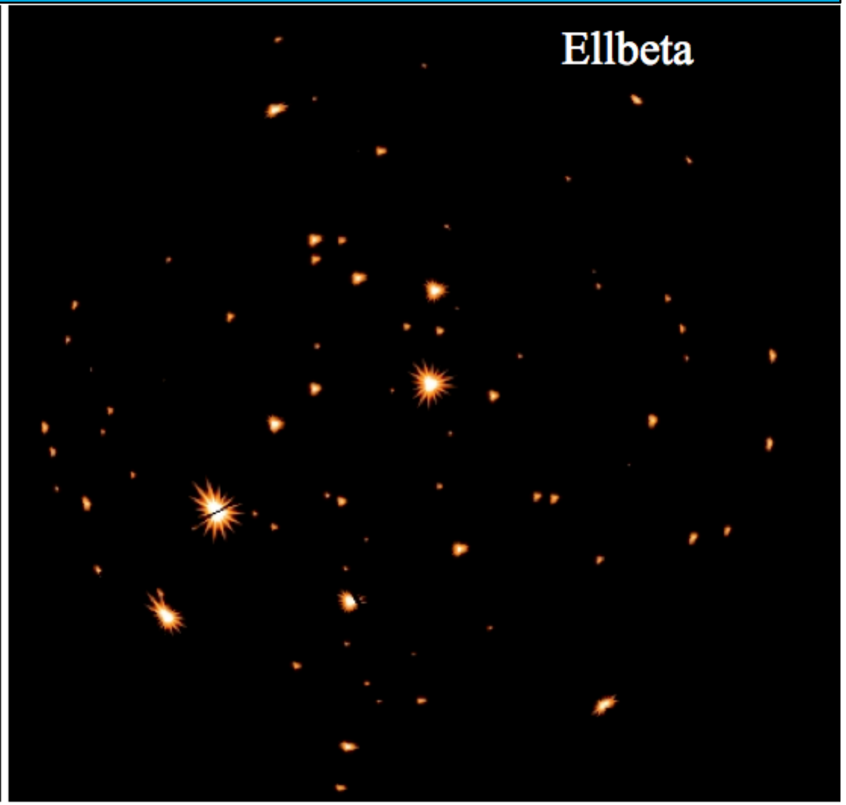
[4] Rotate to correct source phi [5] Az-filter spoke structure [6] Az-filter gross azimuthals

2D PSF in source detection

Source detection running the 2D-PSF model



Data



Model

Issues

Main problems for source detection so far:

- **spurious detections** near bright point sources
  Problem: deviation of CAL PSF models from true EPIC PSF (it was affecting ~ 25% of ext. sources)
  >> much better with 2D-PSF >> SAS 12 >> 3XMM catalogue

- **confusion** of point sources
  Problem: usually a problem by close faint point sources, or by more than 2 close bright sources
  >> ??

- insufficient **background subtraction**
  Problem: limitations of spline fit (eg. by OOT features of pile-up affected sources, etc)
  >> improvement of the background modelling needs calibration of detector induced background features

- **multiple detections** of extended sources
  Problem: β model too simplistic - emldetect tends to add additional sources to the wings
  >> more sophisticated extent model for brighter sources should help

Despite all achievements, there is room for improvements - calibration and s/w people continue working for you