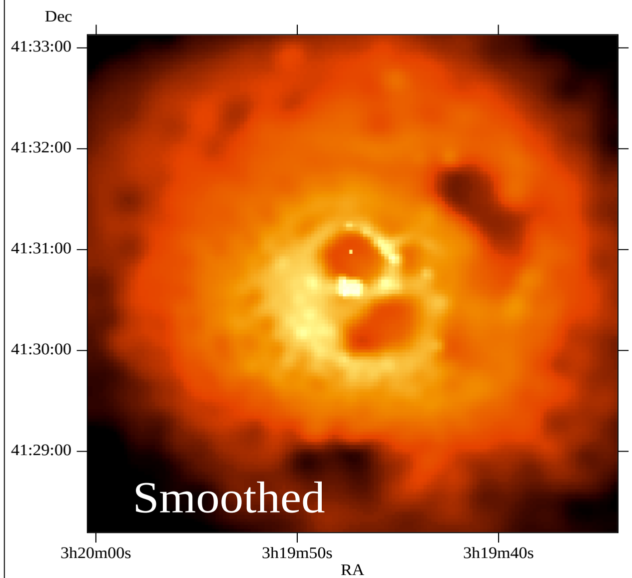
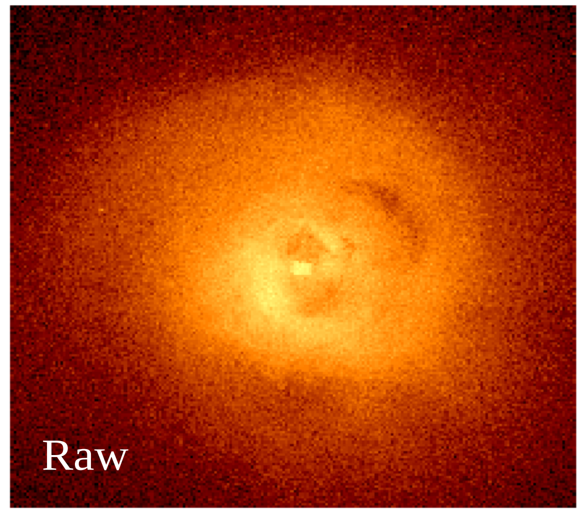
Imaging Analysis

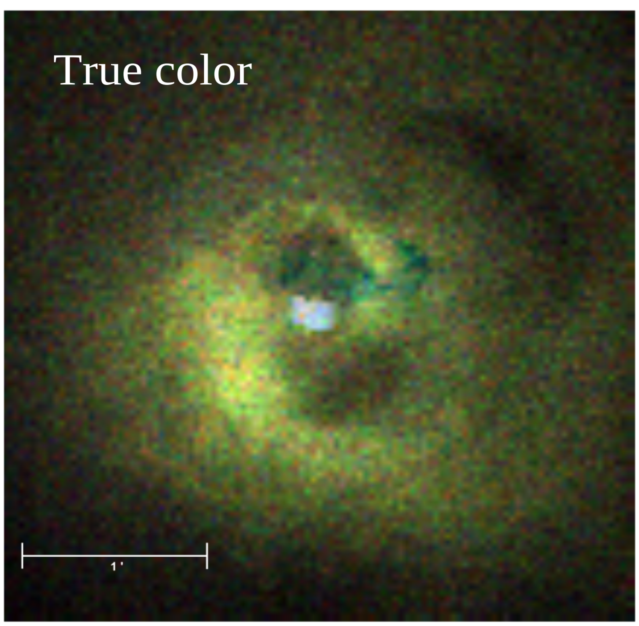
Aneta Siemiginowska
Chandra X-ray Center

Antonella Fruscione and Jonathan McDowell
provided several figures for this talk.

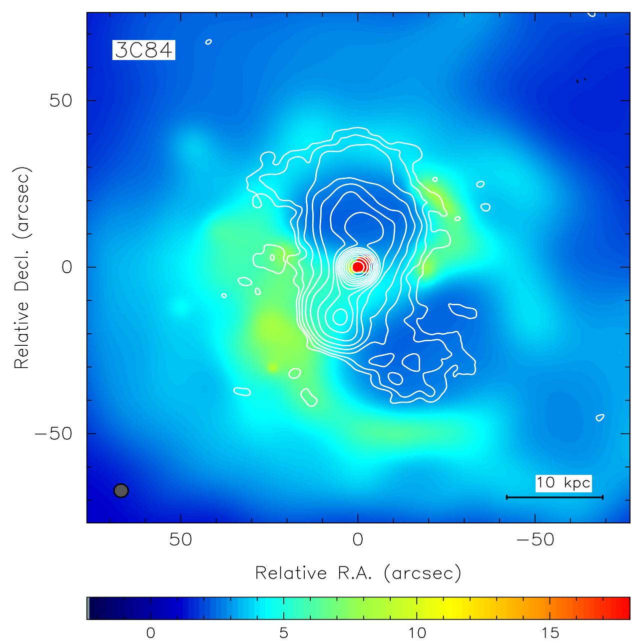
X-ray Images







Perseus A
Chandra ACIS−S



Radio/X−rays

Fabian et al (2000)

- **Intensity maps:**

  colors represent variations in the intensity

- **Raw vs. Smoothed image**

  true counts per pixel

  "average" counts in a pixel

- **True color images:**

  color represents different energy band:

  0.1-0.5 keV red

  1-2 keV green

  2-7 keV blue

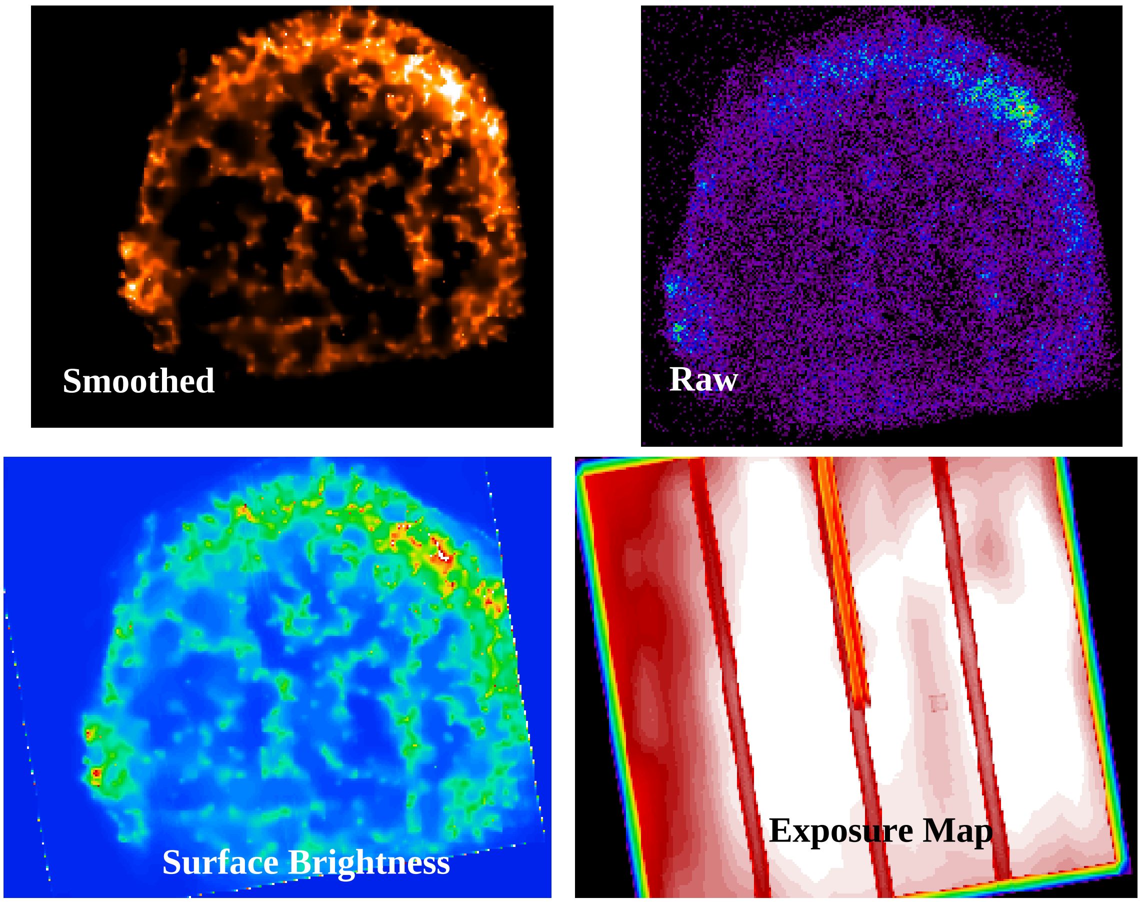
- **Temperature maps:**

  color represents the temperature

- **Images from different part of broad-band emission:**

  radio/X-ray/optical overlays

Counts and Surface Brightness Images



Credit: Holly Jessop

What are the goals of image analysis?

- Create a nice picture.
- Understand the nature of the source:

  Understand the shape and size of emitting regions.

  Understand the temperature distribution, velocity, composition and metallicity etc.

  Differentiate between emission processes.

  Understand energy and power involved in the observed emission.
- Evolution of the source and how it relates to other sources.

SUMMARY

- I will use CIAO software in image analysis.
  (but see IRAF, FTOOLS, XIMAGE, XSPEC)
- Difference between Image and the Event file?
- Binning options
- Display data in different coordinates, detector vs. sky
- Understanding the instrument.
- Detecting sources
  - building the source list for further spectral analysis
  - excluding the sources for the extended source analysis
- PSF effects
- Radial Profile
- 2D fitting in Sherpa
- Smoothing the image
- Image Reconstruction and Deconvolution

EVENT FILES

```
dmlist acis_evt2.fits subspace |more

--------------------------------------------------------------------------------
Data subspace for block EVENTS: Components: 1 Descriptors: 15
--------------------------------------------------------------------------------

 --- Component 1 ---
   1 time                 Real8              TABLE GTI7

                                              68737279.5300000012: 68748578.030
0000012
   2 ccd_id               Int2               7:7
   3 node_id              Int2               0:3
   4 expno                Int4               0:2147483647
   5 chip                 [ 1] chipx         1:1024
   5 chip                 [ 2] chipy         1:1024
   6 tdet                 [ 1] tdetx         1:8192
   6 tdet                 [ 2] tdety         1:8192
   7 det                  [ 1] detx              0.50:     8192.50
   7 det                  [ 2] dety              0.50:     8192.50
   8 sky                  [ 1] x                 0.50:     8192.50
   8 sky                  [ 2] y                 0.50:     8192.50
   9 pha                  Int4               0:36855
  10 energy               Real4              0:  1000000.0
  11 pi                   Int4               1:1024
  12 fltgrade             Int2               0:255
  13 grade                Int2               0:0,2:2,3:3,4:4,6:6
  14 status               Bit
  15 phas                 Int2               -4096:4095
```

CHANDRA Coordinates Systems

- **SKY(X,Y)** gives pixels in the tangent plane to a specified position on the celestial sphere. Tangent point is in the center of the image, for ACIS it is (4096.5, 4096.5). Header information lets `dmcoords` and `DS9` convert these pixel numbers to celestial RA and Dec (J2000).

  ```
  dmcopy ``evt.fits[bin x=3500:4500:4,y=3500:4500:4]''
  sky_by_4.img
  ```

- **CHIP(CHIPX,CHIPY)** gives row and column number on each chip.

  If you make an image in chip coordinates, you should first filter on CCD_ID to pick out a single chip:

  ```
  dmcopy "evt.fits[ccd_id=7][bin chip]" chip7.img
  ```

- **TDET(TDETX,TDETY)** "Tiled Detector" coordinates. The chips placed next to each other

  ```
  dmcopy "evt.fits[bin tdet]" tdet.img
  ```

  NOTE: offsets between the chips are arbitrary. TDET should be used for visualization only.

- **DET(DETX,DETY)** "Detector coordinates" a projection of the photon positions onto the tangent plane to the unit sphere, with the tangent point being the Chandra telescope's optical axis. The center of the coordinate system is (4096,4096) for ACIS.

```
dmcopy "evt.fits[bin det=16]" det_by_16.img; ds9 det_by_16.img
```

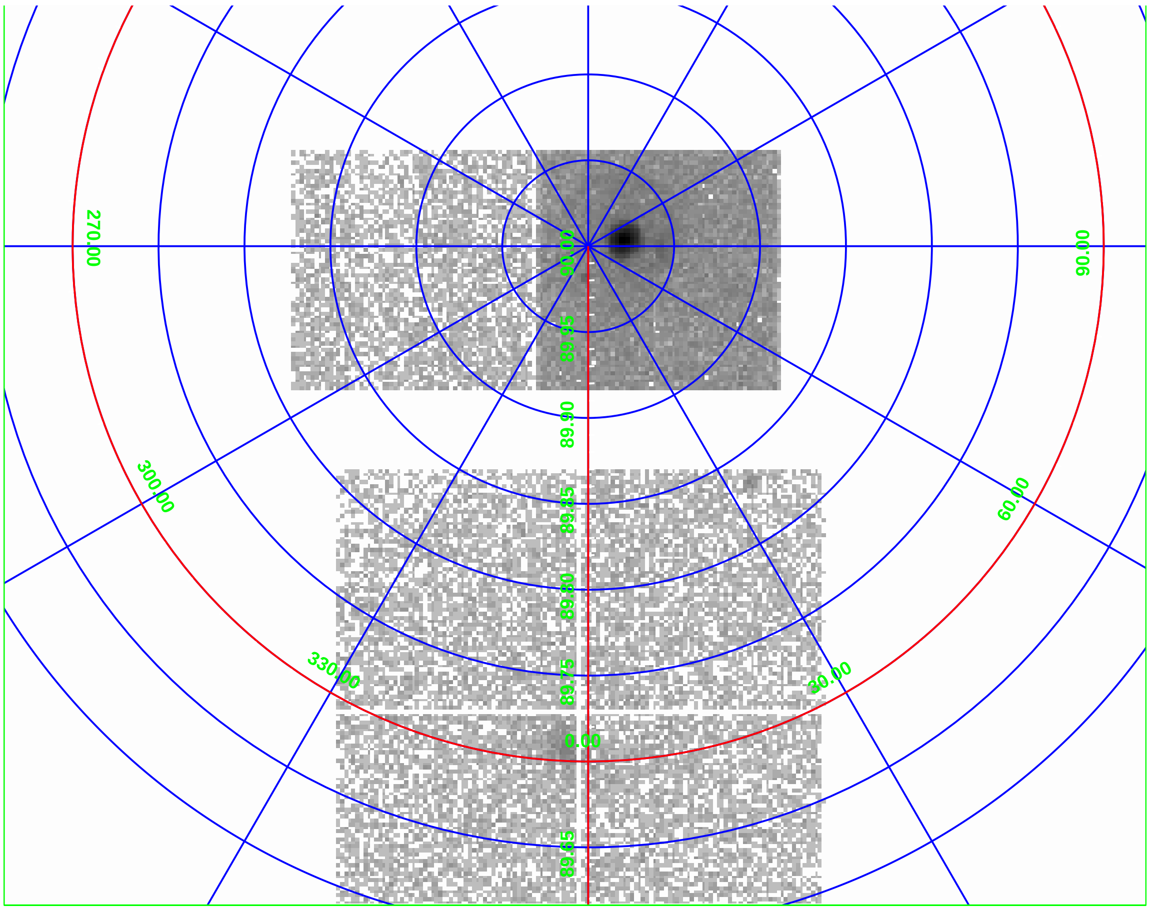
or ds9 "evt.fits[bin=detx,dety]"

o) set Analysis/Coordinate Grid Parameters/Coordinate to "degrees"

o) set Edit/Preferences/WCS to "degrees"

o) set Analysis/Display Coordinate Grid

You will then see a grid marking the off-axis angles, making it easy to read off the values. The numbers in the WCS window are azimuth and off-axis angle; the numbers displayed on the coordinate grid are azimuth and (90 minus off-axis angle), so that the grid is marked 90.00 at the pole.


270.00
90.00
300.00
60.00
330.00
30.00
0.00
90.00
89.95
89.90
89.85
89.80
89.75
89.65

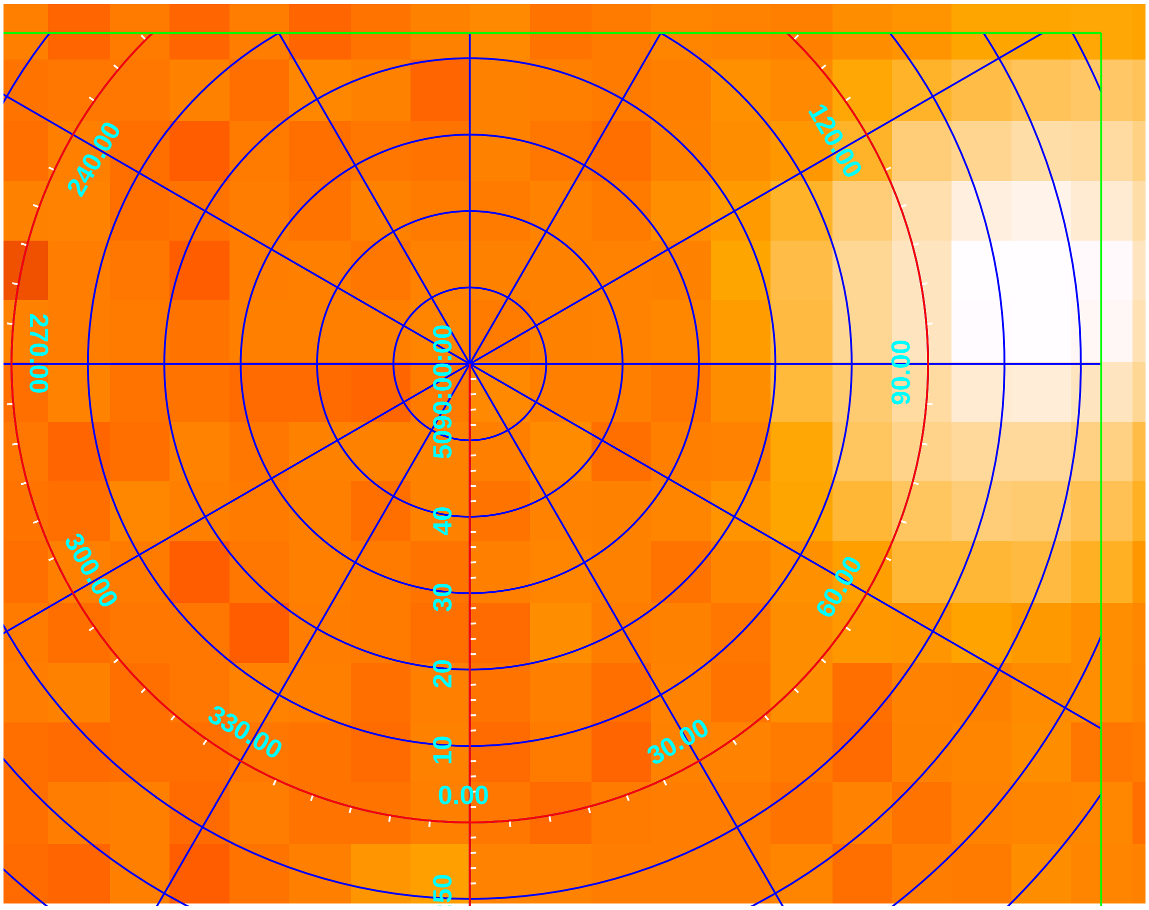


Image in Detector coordinates: zoomed

```
miles-126: dmcoords acis_evt2.fits
dmcoords>: sky 4250 4112
(RA,Dec):      18:33:27.528    -10:35:16.05
(RA,Dec):       278.36470      -10.58779 deg
THETA,PHI           1.265'        272.57 deg
(Logical):        4250.00        4112.00
SKY(X,Y):         4250.00        4112.00
DETX,DETY         4103.43        3942.38
CHIP ACIS-S3       227.64         563.95
TDET              4144.64        2265.95
```

BINNING EVENTS INTO IMAGES

- In CIAO the event files can be **binned** and **filtered** with `dmcopy`.
- Images can be created in the standard spatial coordinates (x,y).

```
dmcopy ``acis_evt2.fits[bin x=3200:4200:2, y=3200:4200:2]''
acis_img.fits
dmcopy ``acis_evt2.fits[bin x=8, y=8]'' acis_img_by8.fits
```

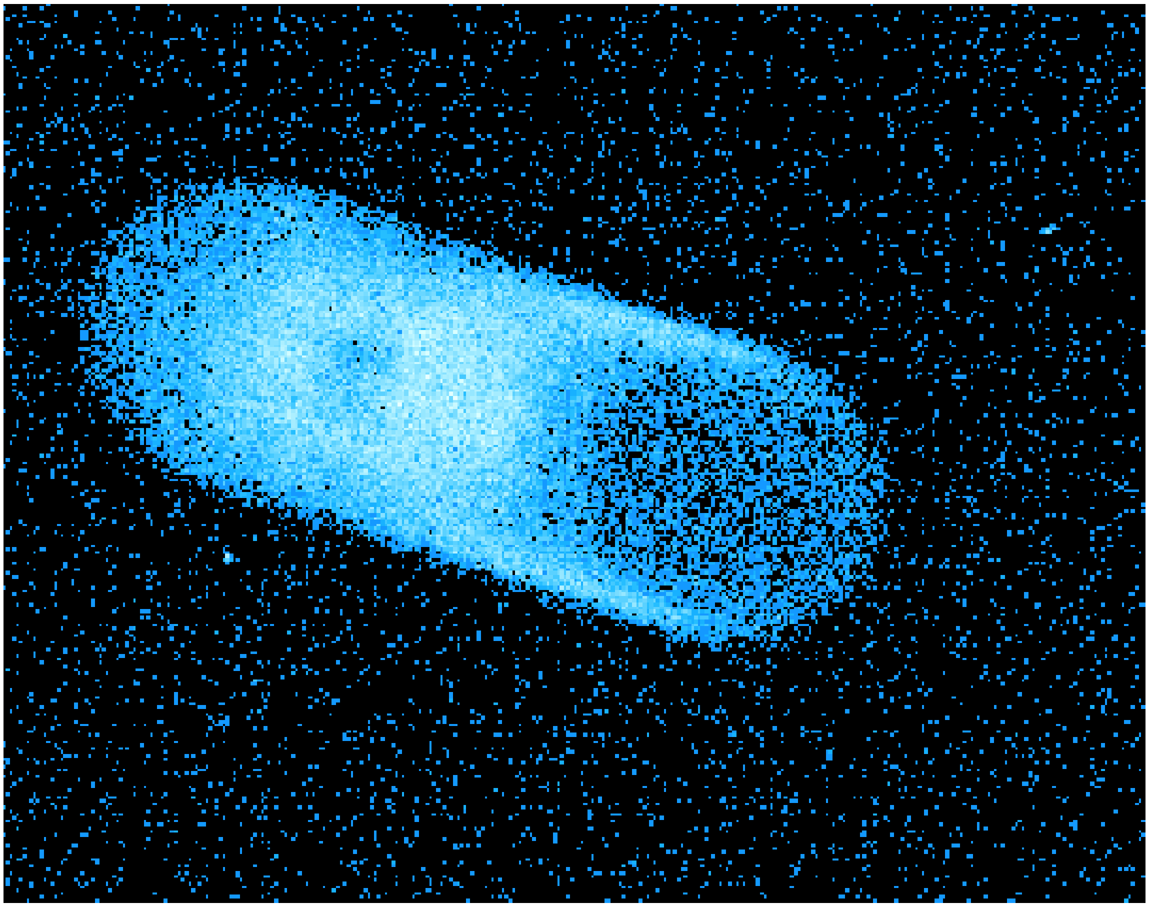
- Images can be also created in different coordinates and dimensions e.g. (energy,time), (energy,x), (energy,y), (energy, x, y) (time, x y).

```
dmcopy ``acis_evt2.fits[bin y=3200:4200:2, energy=300:7500:50]''
acis_y_E_img.fits

dmcopy "acisf01463N002_evt2.fits[bin time=59969680.7:59992061.4:200,
y=4100:4400:2]" time_y_img.fits

dmcopy "acisf01463N002_evt2.fits[bin x=3900:4400:2,y=4100:4400:2,
time=59969700:59992000:1000]" time_x_y_img3d.fits
```

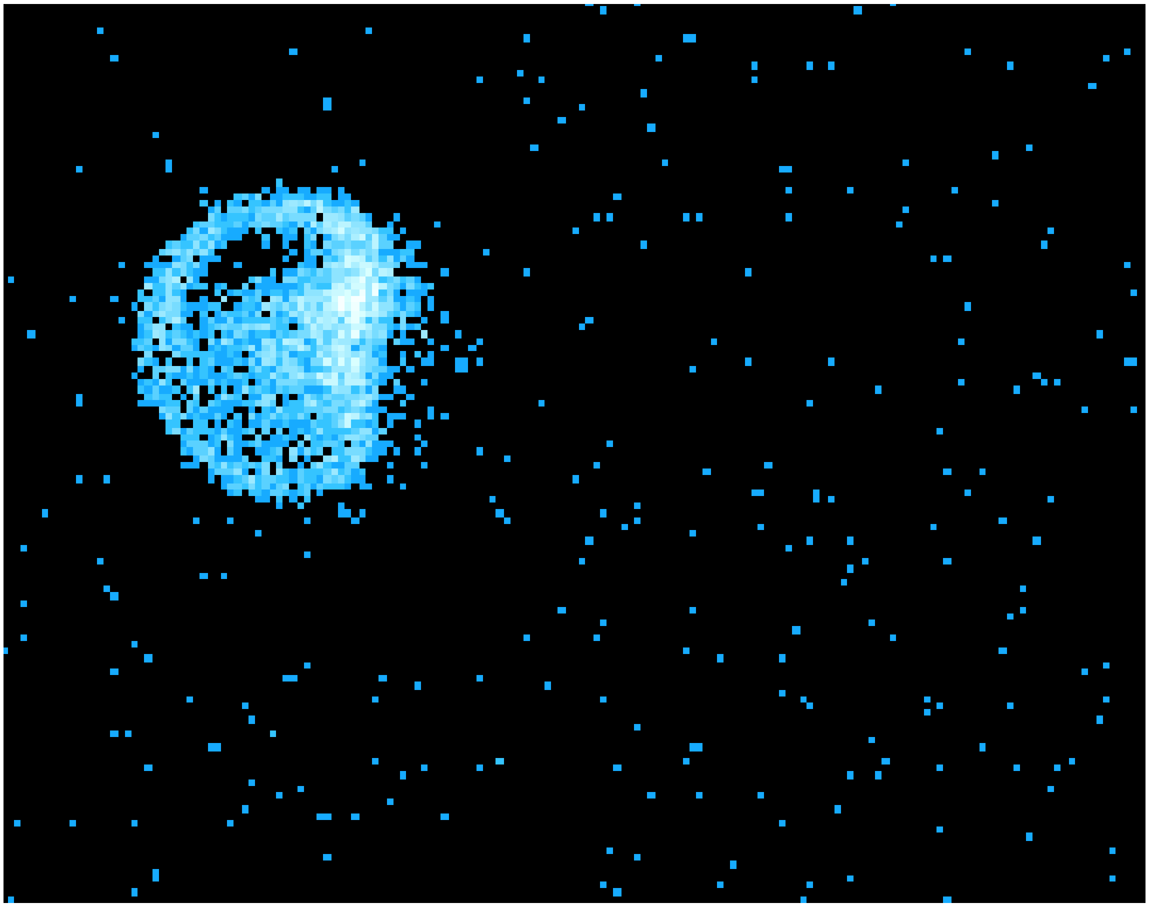
- Filtering on energy to create images in different bands.

```
dmcopy ``acis_evt2.fits[energy=300:2500][bin
x=3200:4200:2,y=3200:4200:2]'' acis_img_0.3-2.5keV.fits
```

- True Color Images **`CIAO tools: dmcopy, dmimg2jpeg`**



Position vs. time



Position: sky(x,y)

INSTRUMENTAL FEATURES

- Understanding the instrument:

  CCD is different than microchannel plate.

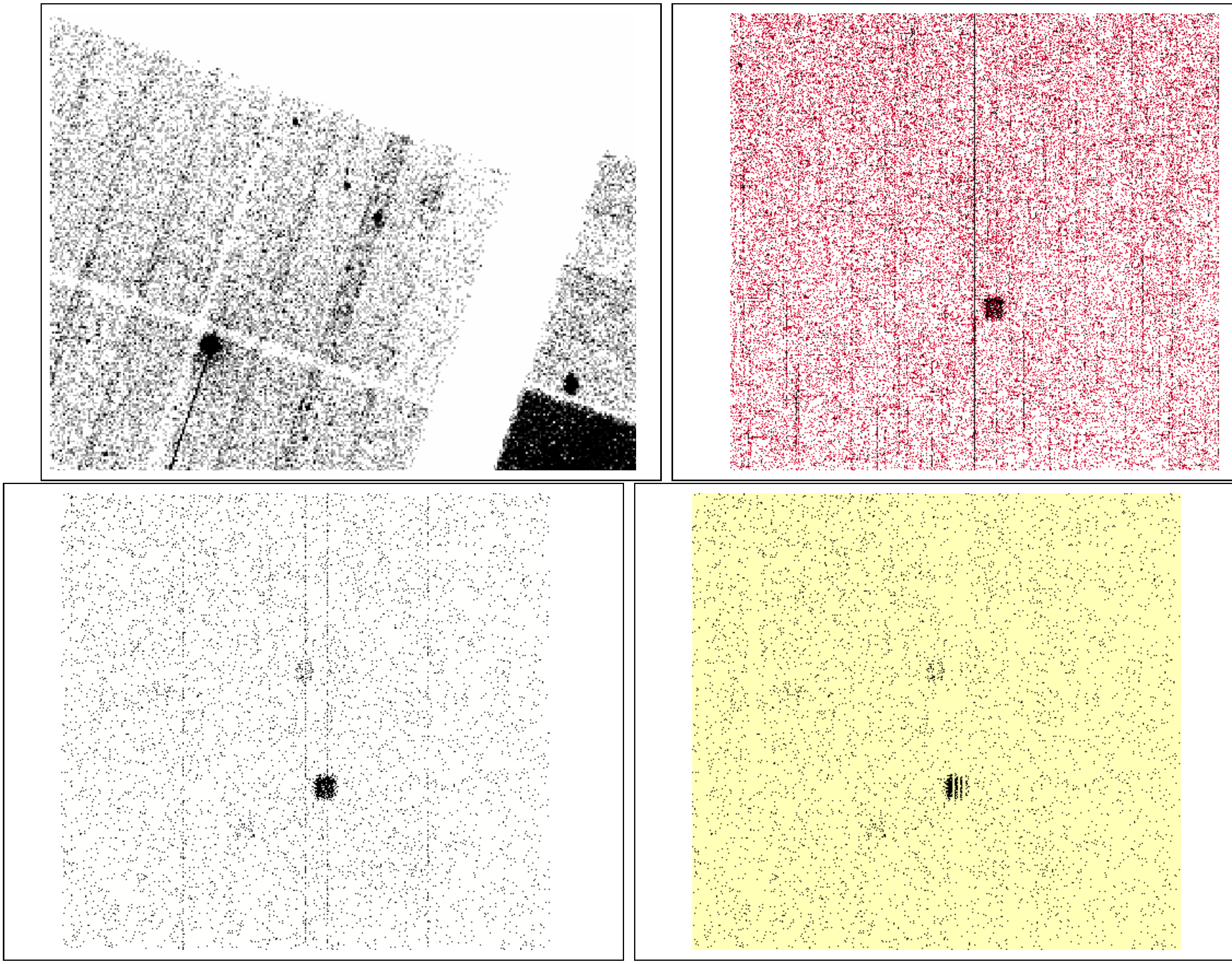
- Bad pixels and columns:

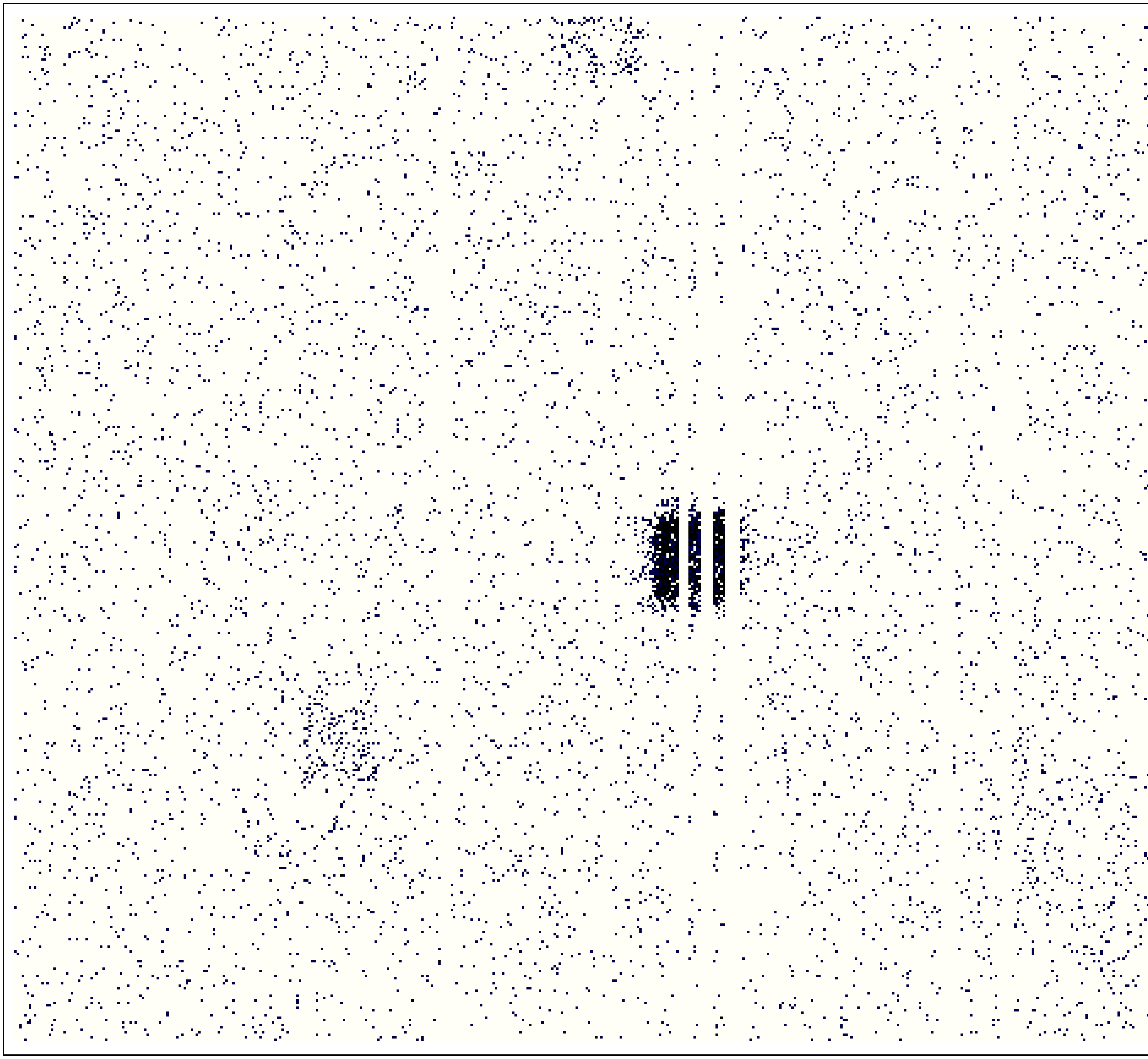
  hot pixels, node boundaries etc.

  `bpix1` file associated with the observations can be used to remove bad pixels from ACIS event file.

- Trailed images (Out of Time Events).

- Pile-up.





CHIP 8 (S4) STREAKS

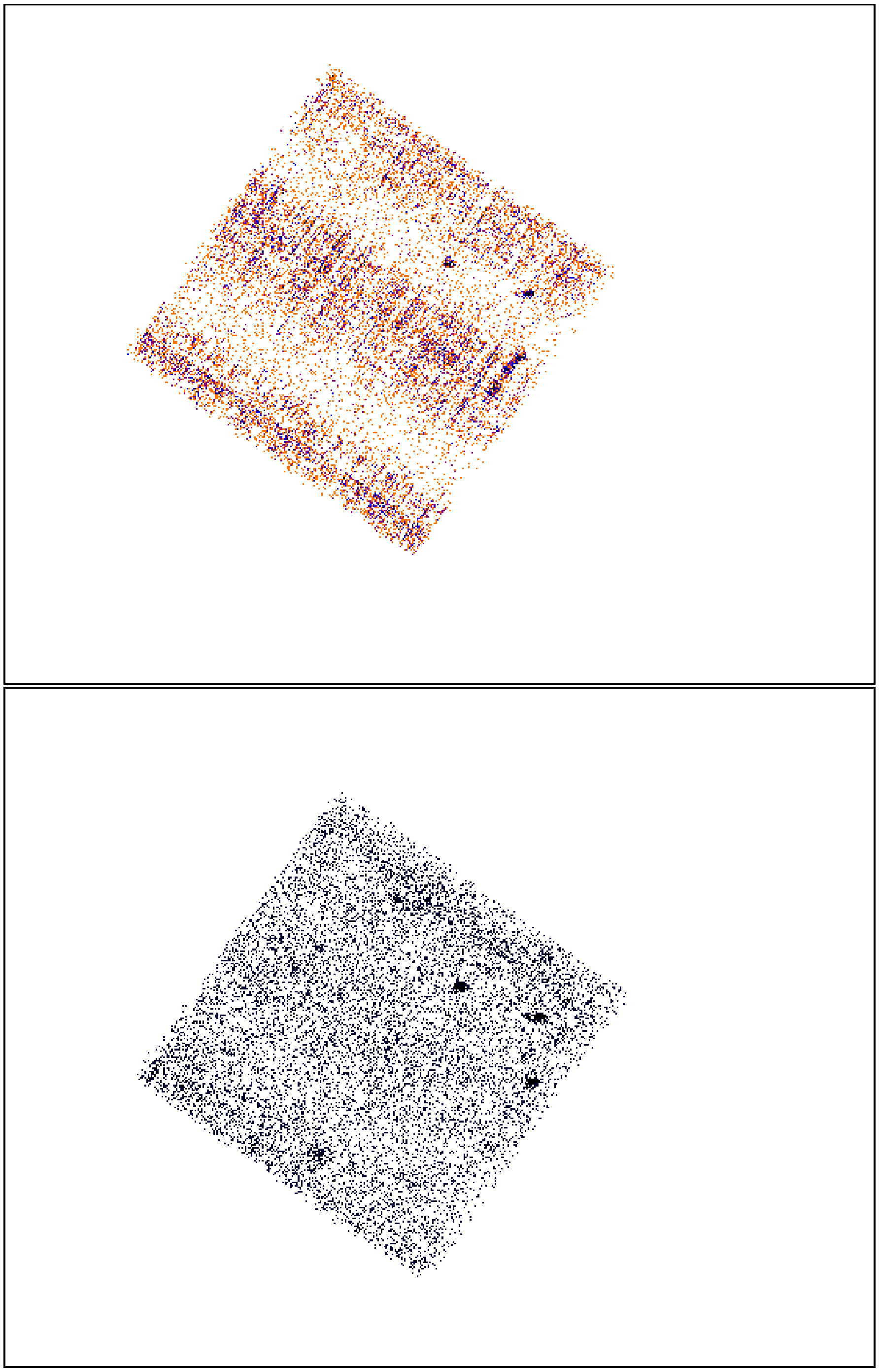
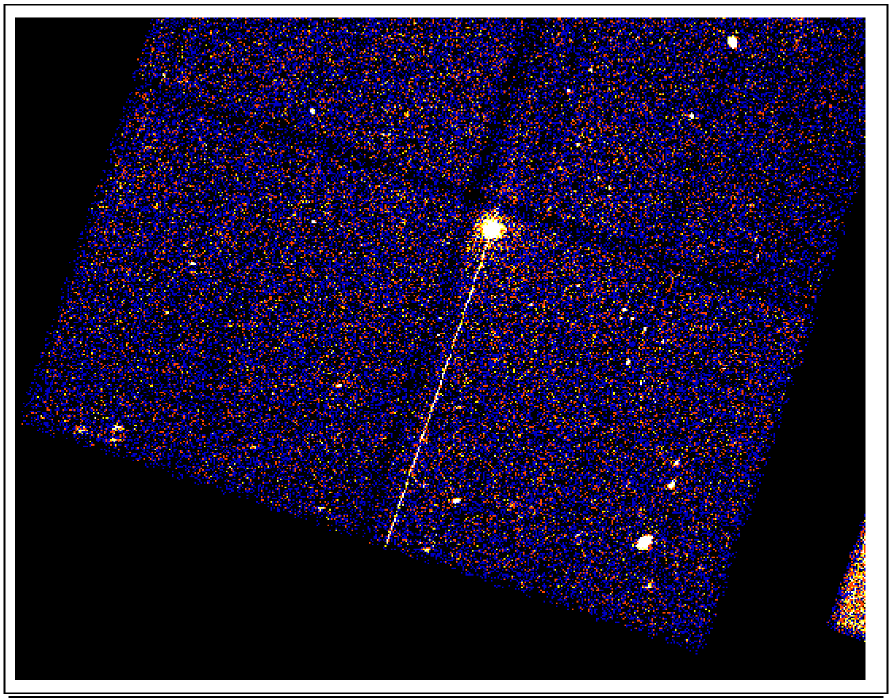


Figure 3: WITH AND WITHOUT HOUCK ALGORITHM

OUT-OF-TIME EVENTS



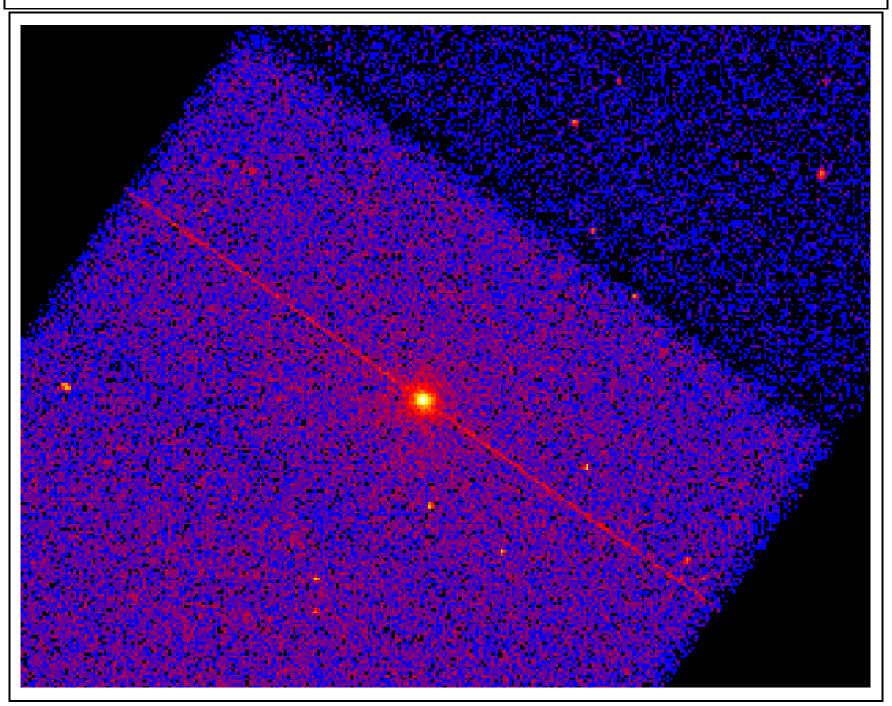


Figure 4: BEFORE CORRECTION

OUT-OF-TIME EVENTS

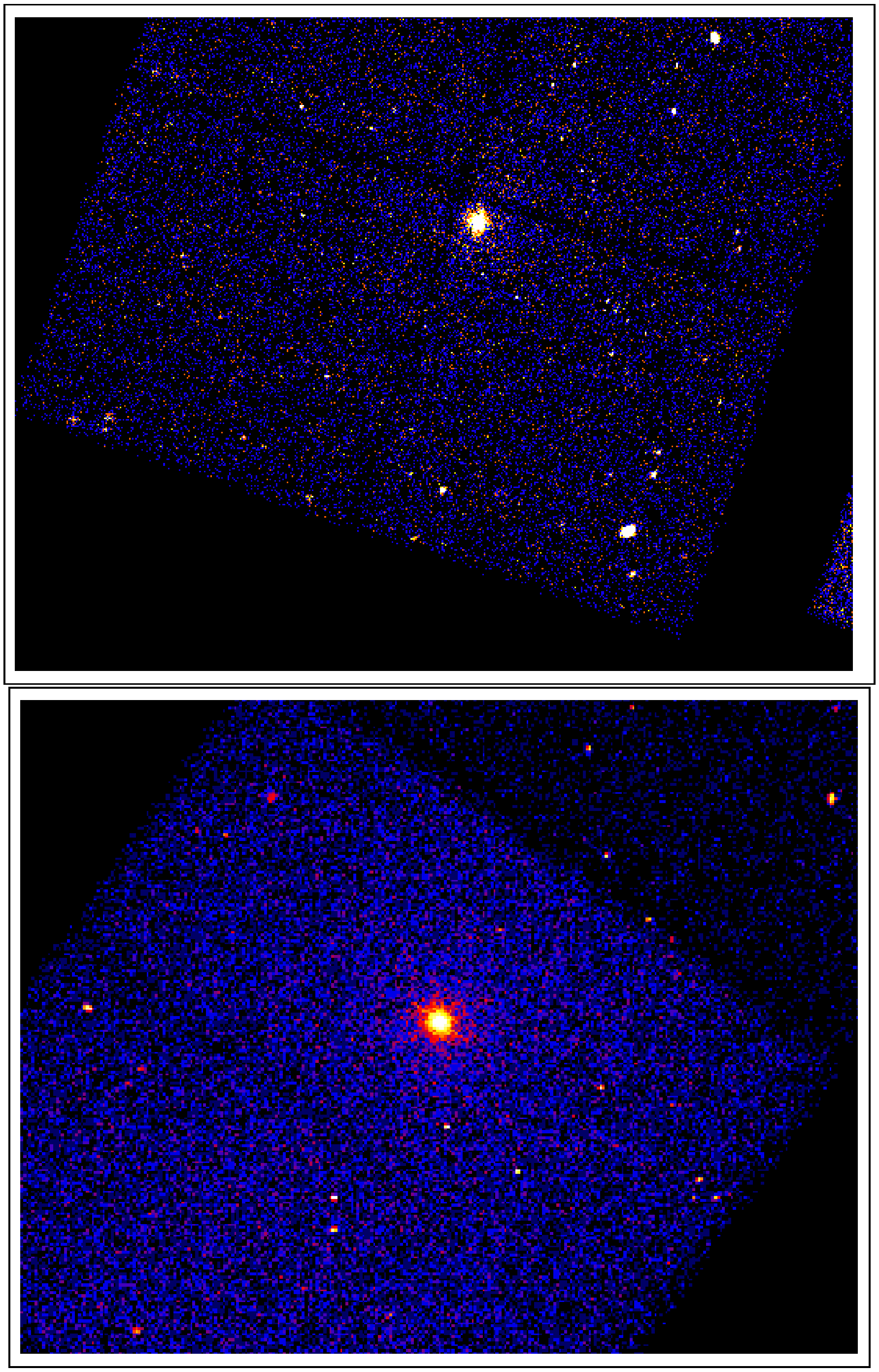


Figure 5: AFTER CORRECTION WITH ACISREADCORR

INSTRUMENT CHARACTERISTICS

- Instrument Map (`CIAO tool: mkinstmap`)

  an **image** in detector coordinates which is the product of the mirror effective area projected onto the detector surface with the QE of the detector. It incorporates bad pixels and ACIS windows (if any) by setting the efficiency to zero at the appropriate locations on the detector. [cm^2 counts]

- Exposure Map (`CIAO tool: mkexpmap`)

  an **image of the effective area** at each sky position, accounting for the effects of dither [cm^2 counts]

- Background Map (`lc_clean`)

  scripts in CIAO 2.2 `make_acisbg`

- Point Spread Function (**PSF**)(`CIAO tool: mkpsf`)

  Image produced by a delta function (point) source on the detector. ('Point Response Function' or PRF).

Exposure Map

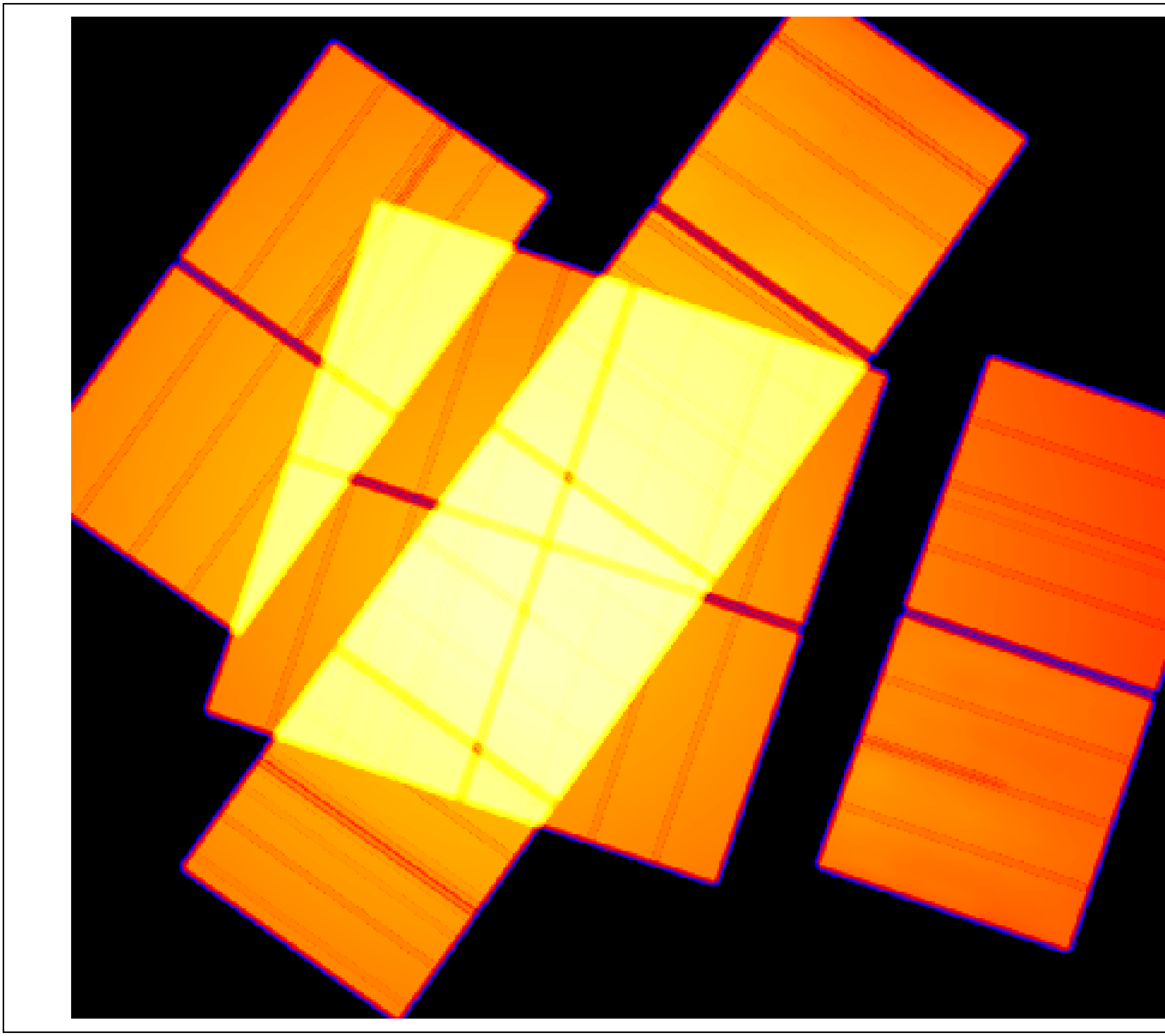


Figure 1: The exposure map allows us to calculate the relative exposure at any pixel. It takes into account energy-dependent sensitivity variations.

Exposure Map Energy Dependence

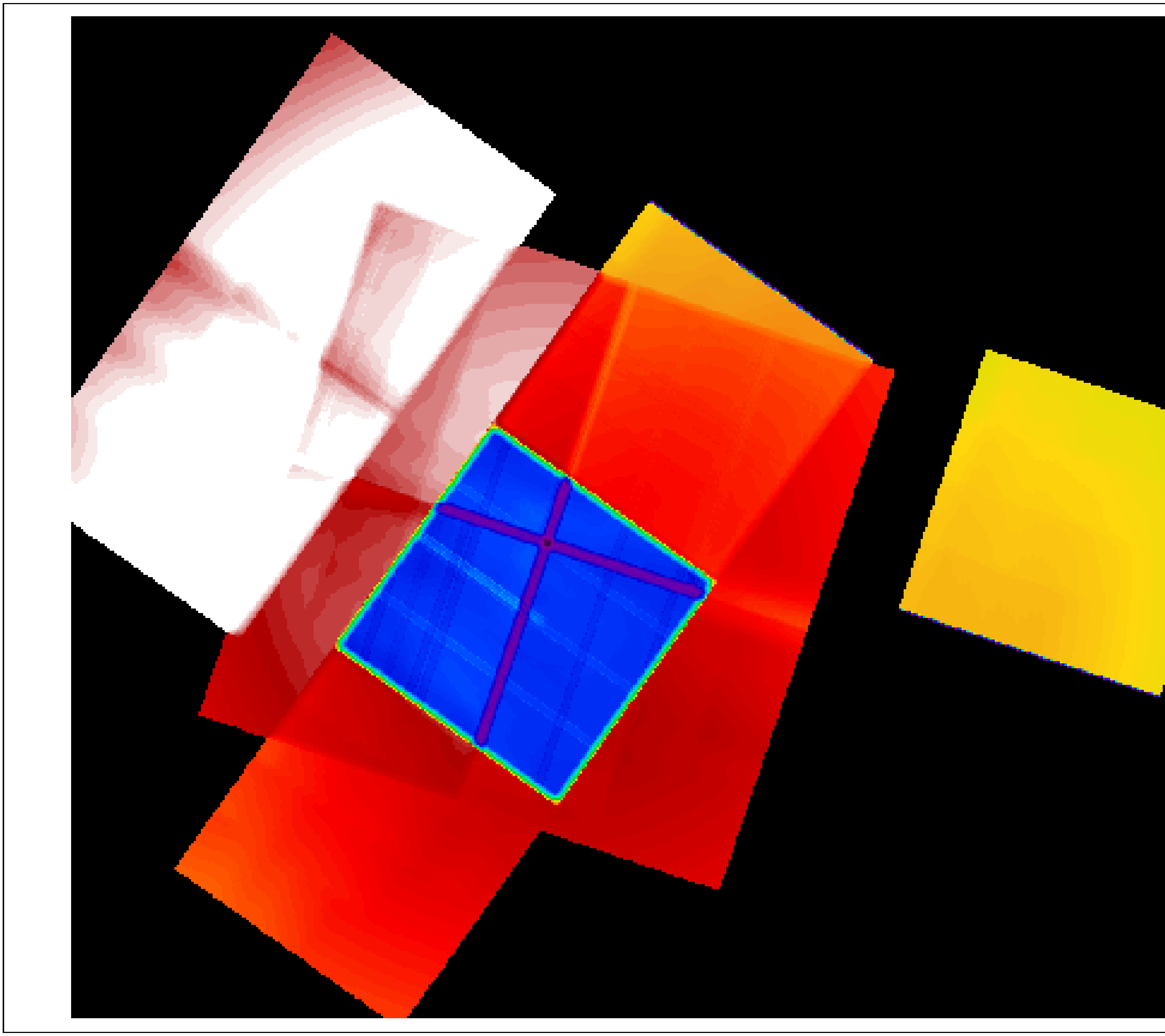


Figure 2: The exposure map varies spatially with energy. However, the variations within a chip are much smaller than the variations from chip to chip.

Background Map

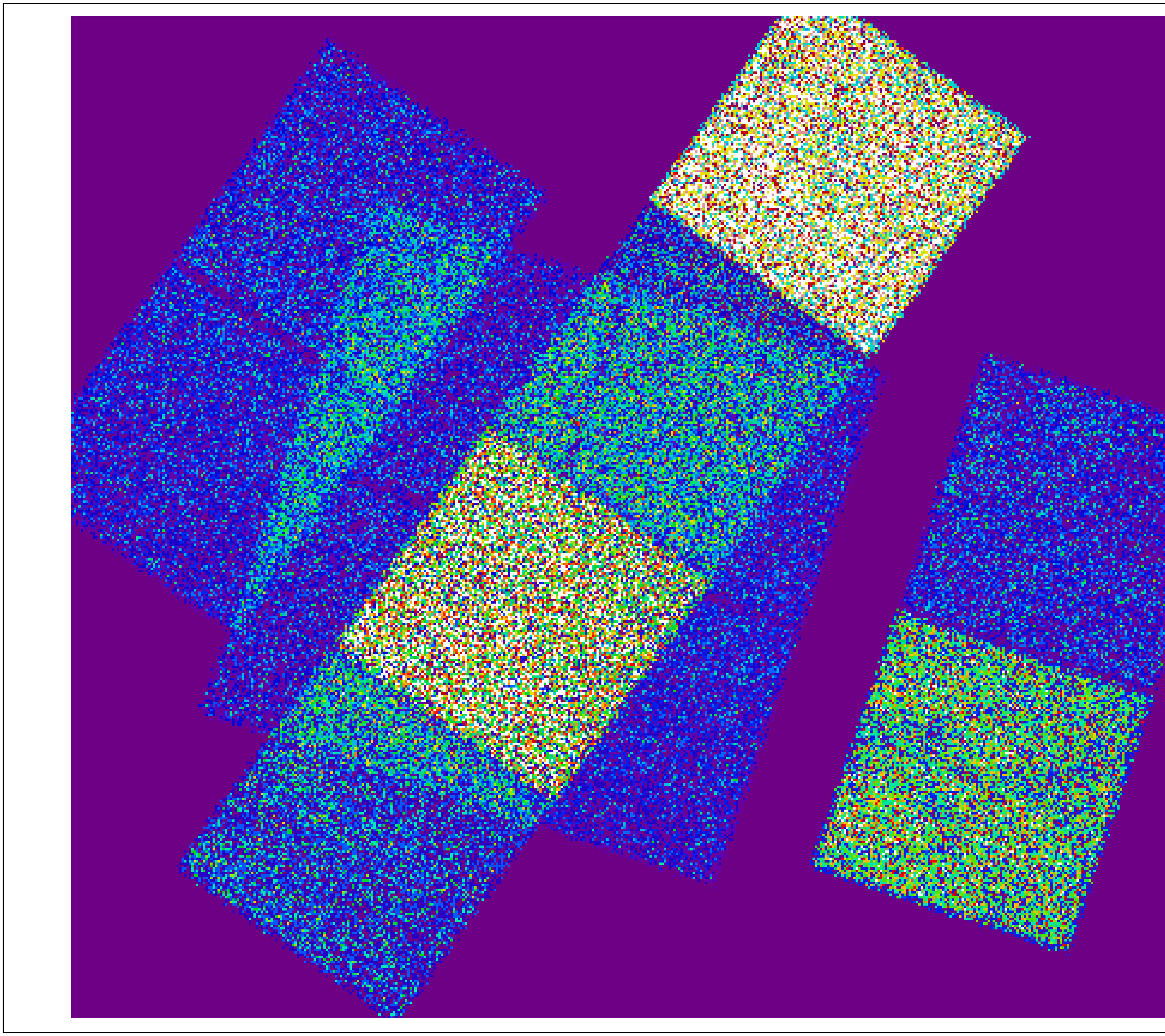
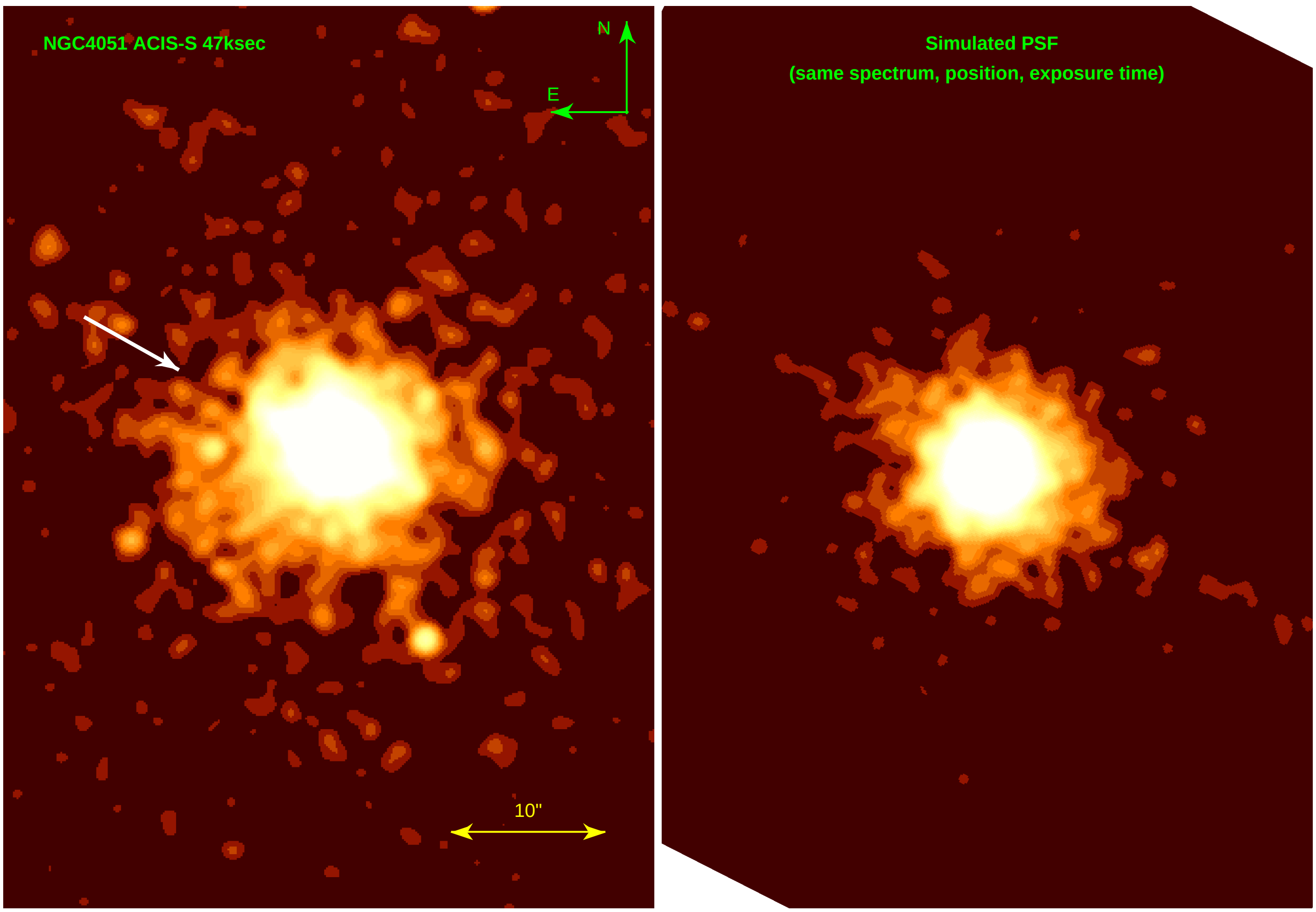


Figure 3: The background map is made using Maxim Markevitch's background event files. The contributions have been adjusted manually on a per-chip basis to ensure a zero mean in source-free areas of the background-subtracted image. In particular, the BI chip backgrounds are stronger than the standard quiescent values.


NGC4051 ACIS-S 47ksec
N
E
10"
Simulated PSF
(same spectrum, position, exposure time)

Source Detection

- **Detect algorithms in CIAO:**

  * **Celldetect:**

  - basic, simplest detection method
  - convolves 2D image with the square cells filter
  - compares the counts in the source and background cells
  - the cell size is determined by the size of the PSF
  - uses the threshold to record candidate source

  * **Wavdetect**

  - convolves the binned 2D image with the wavelet function at different scales.
  - uses "Mexican Hat" wavelet
  - Background is estimated from the "cleansed" image and used to set detection thresholds

  Note: threshold depends on scale

  * **VTPdetect**

  - works directly on the event file (but also accepts the image)
  - locates groups of photons (using Delauney triangulation and Voronoi tesselation)
  - groups each set of photons into a source
  - works on low surface brightness images
  - detect the point sources and SNR shells alike structures
  - independent on shape and geometry

CHANDRA SOURCE DETECTIONS

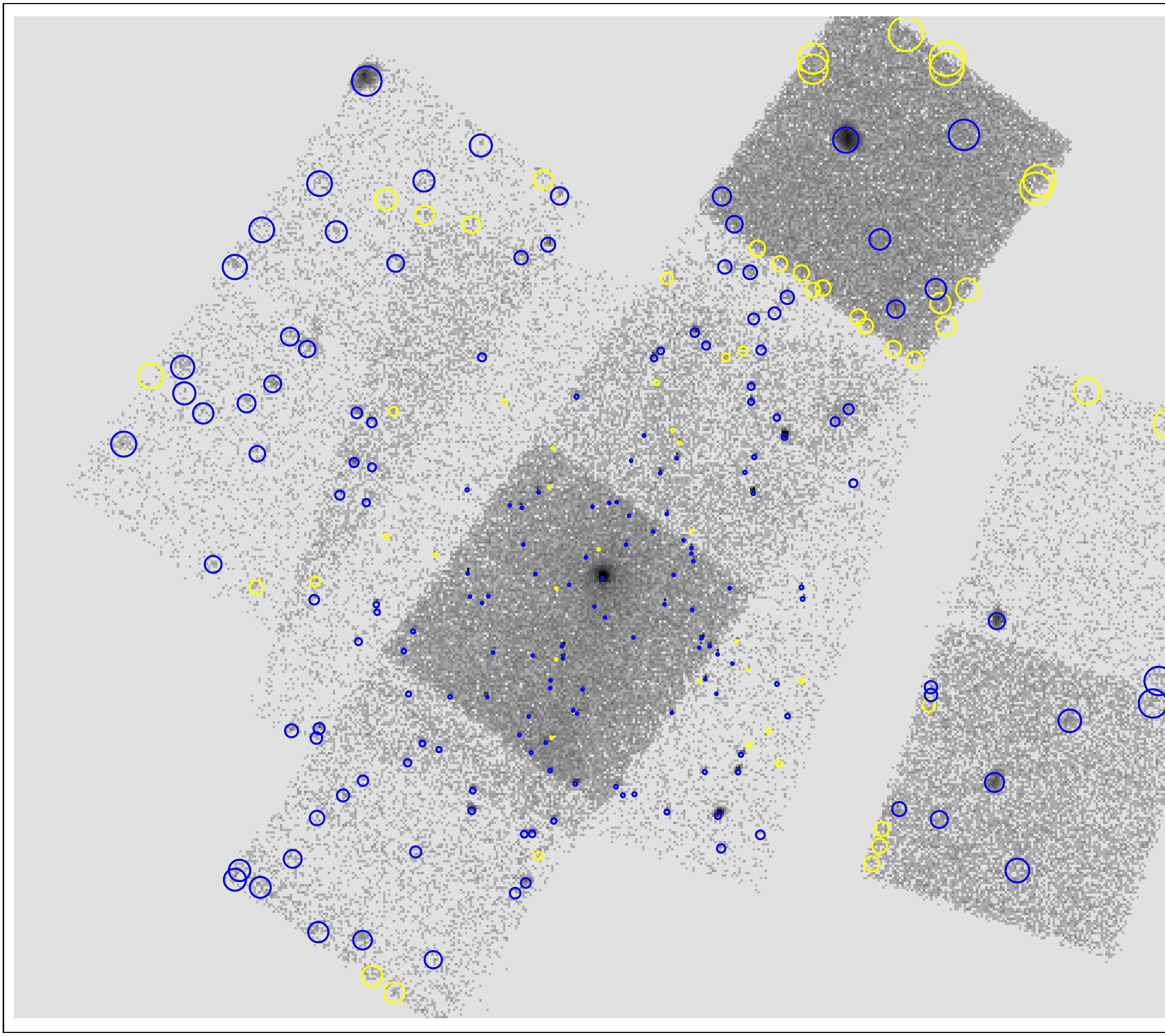


Figure 8: Sources identified by the CIAO wavelet detection package. Circle size increases with PSF size. A total of 170 sources (at a false source probability level of 10^{-6}) were found, after excluding spurious sources. Sources marked in yellow have been removed as lower significance or spurious - the algorithm has problems along the BI/FI chip boundaries where the background changes abruptly, and at the chip edges. Otherwise the wavelet algorithm has done a good job in detecting real sources. The algorithm was applied to the raw count image, with the exposure map used for thresholding (without which the program would detect large numbers of spurious sources at the chip edges).

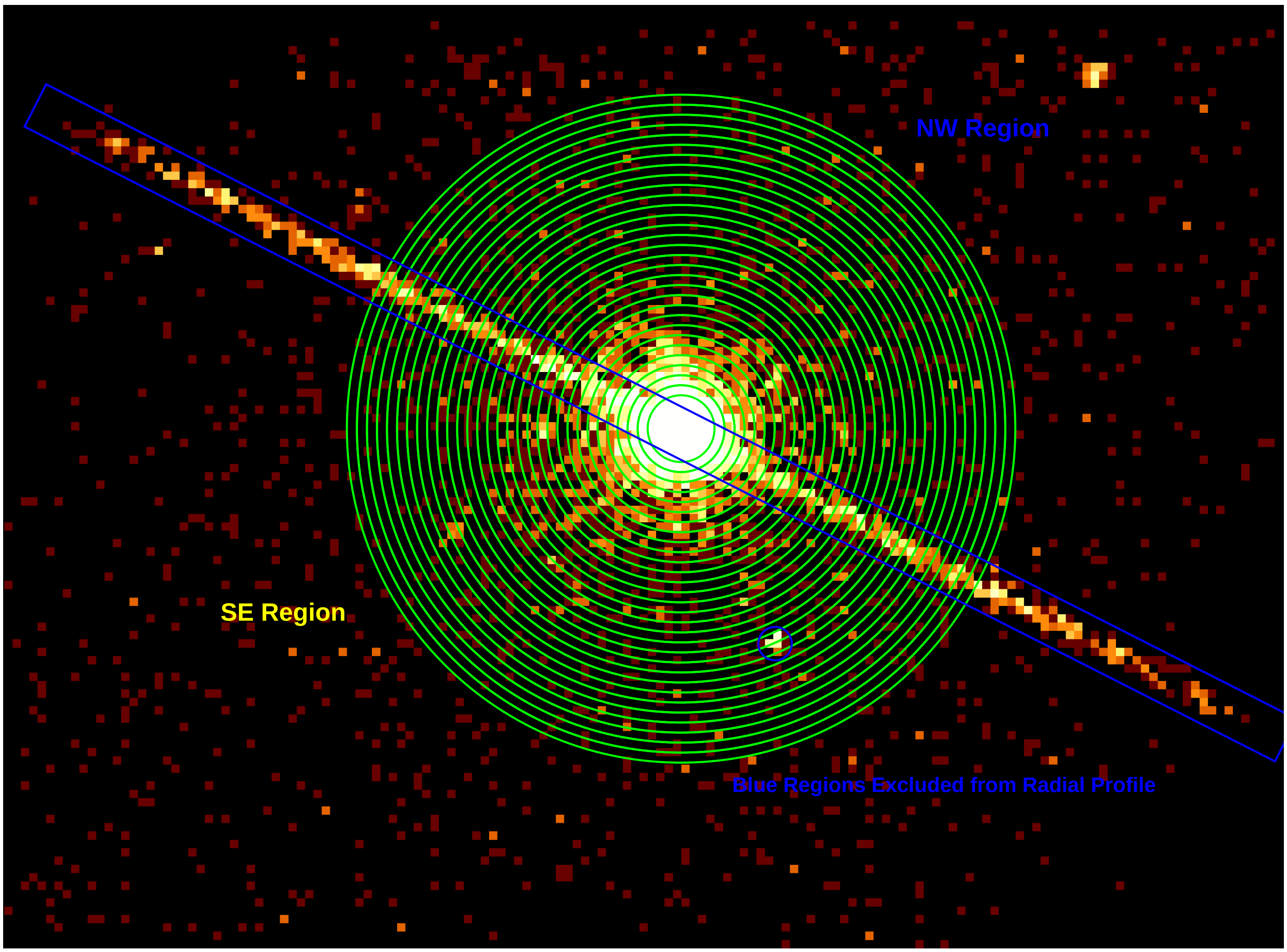
Radial Profile Analysis

- Preparation:
  - Determine the overall mean spectrum of the source: **dmextract, mkrmf, mkarf, Sherpa or XSPEC**
  - Create exposure map using this model as a weighting function **mkexpmap**
  - Make a set of images with the same size and binning as the exposure maps: **dmcopy**
  - Detect point sources: **WAVDETECT**
  - Exclude all point sources from the exposure map: **dmcopy**
  - Define the regions for radial profile
- `CIAO tool: dmextract`

```
dmextract infile="acis_all_290_7000_img.fits
[bin sky=annulus(4060.5,4225,2:542:60)]"
outfile=profile_exp.fits exp="acis_all_290_7000_filt_emap.fits"
```

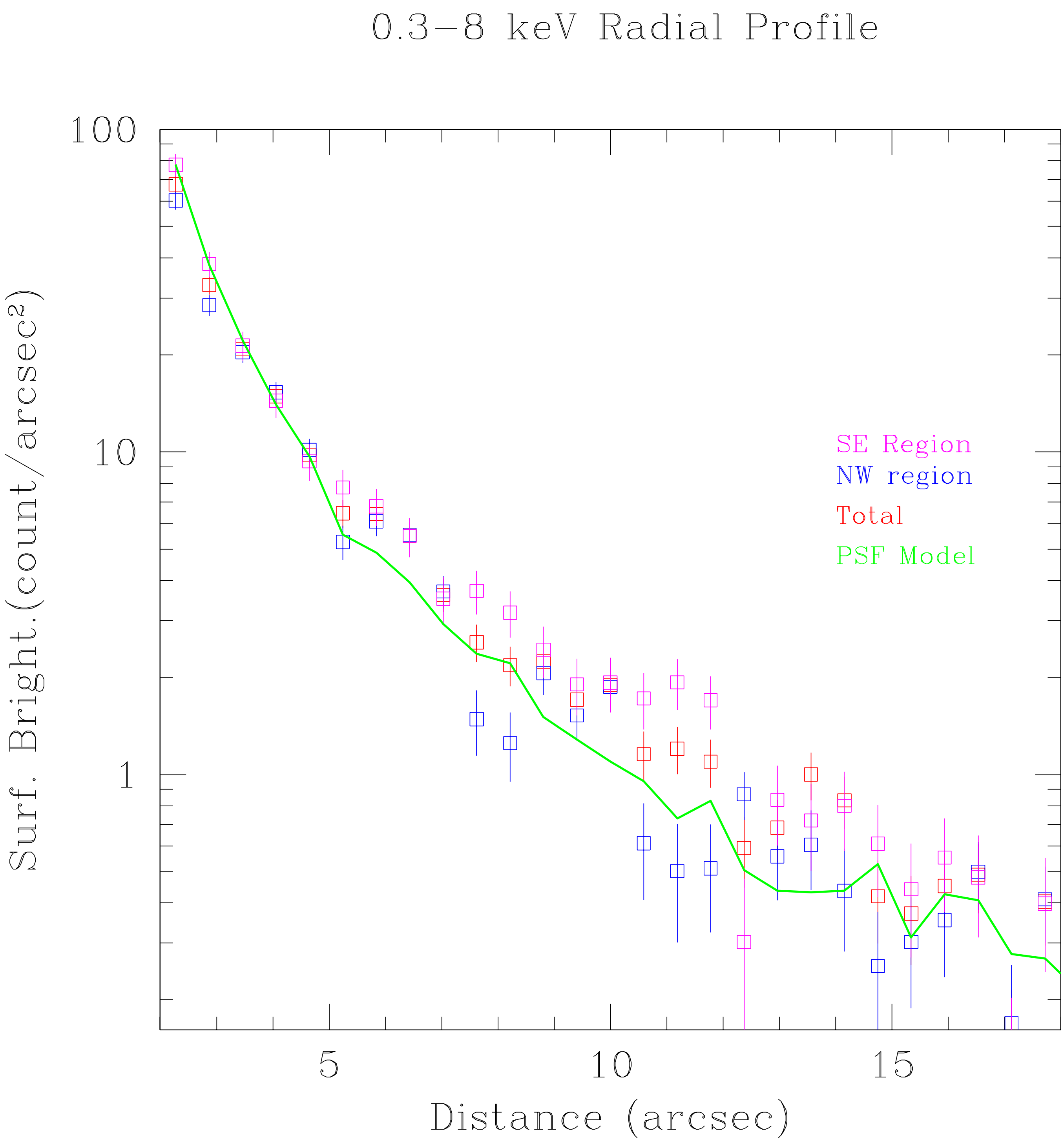
The output file is a FITS file and each row in the file corresponds to one region. There are 15 columns in the file:

```
unix%dmlist profile_big.fits blocks

--------------------------------------------------------------------------------
Dataset: profile.fits
--------------------------------------------------------------------------------

     Block Name                          Type         Dimensions
--------------------------------------------------------------------------------
Block    1: PRIMARY                        Null
Block    2: HISTOGRAM                      Table        17 cols x 15       rows


unix% dmlist profile_big.fits subspace

--------------------------------------------------------------------------------
Data subspace for block HISTOGRAM: Components: 1 Descriptors: 17
--------------------------------------------------------------------------------

 --- Component 1 ---
   1 POS                   [ 1] X                     1864.0:     6264.0
   1 POS                   [ 2] Y                     2019.0:     6419.0
   2 EQPOS                 [ 1] RA              DEFAULT
   2 EQPOS                 [ 2] Dec             DEFAULT
   3 SHAPE                 String
   4 R                     Real8                DEFAULT
   5 ROTANG                Real8                DEFAULT
   6 COMPONENT             Int2                 DEFAULT
   7 AREA                  Real4                DEFAULT
   8 EXPOSURE              Real8                DEFAULT
   9 COUNTS                Real8                DEFAULT
  10 ERR_COUNTS            Real8                DEFAULT
  11 COUNT_RATE            Real8                DEFAULT
  12 NET_COUNTS            Real8                DEFAULT
  13 NET_ERR               Real8                DEFAULT
  14 NET_RATE              Real8                DEFAULT
  15 ERR_RATE              Real8                DEFAULT
  16 SUR_BRI               Real8                DEFAULT
  17 SUR_BRI_ERR           Real8                DEFAULT
```


NW Region
SE Region
Blue Regions Excluded from Radial Profile

- Fitting the radial profile in Sherpa.
  - read the surface brightness profile and the errors
  - define the model - Beta profile:

```
        1-D surface brightness beta-model:
        f(x) = A*(1+[(x-xpos)/r_0]^2)^(-3*beta+1/2)

        BETA1D Parameters:

        1       r0      core radius r_0
        2       beta    beta index
        3       xpos    offset from x = 0
        4       ampl    amplitude A at (x - xpos = 0)


Sherpa commands:
----------------

data "profile_exp0.fits[cols COMPONENT, SUR_BRI]"
read errors "profile_exp0.fits[cols COMPONENT,SUR_BRI_ERR]"
source = const1d[con]+beta1d[b1]
fit

project
        --------------------------------------------------------
        Parameter Name      Best-Fit Lower Bound     Upper Bound
        --------------------------------------------------------
           con.c0       2.56761e-05  -4.52182e-07    +4.43959e-07
            b1.r0           6.09924  -0.108983       +0.108506
            b1.beta        0.577214  -0.00268654     +0.00268394
            b1.ampl       0.0407426  -0.00079724     +0.000836057
```

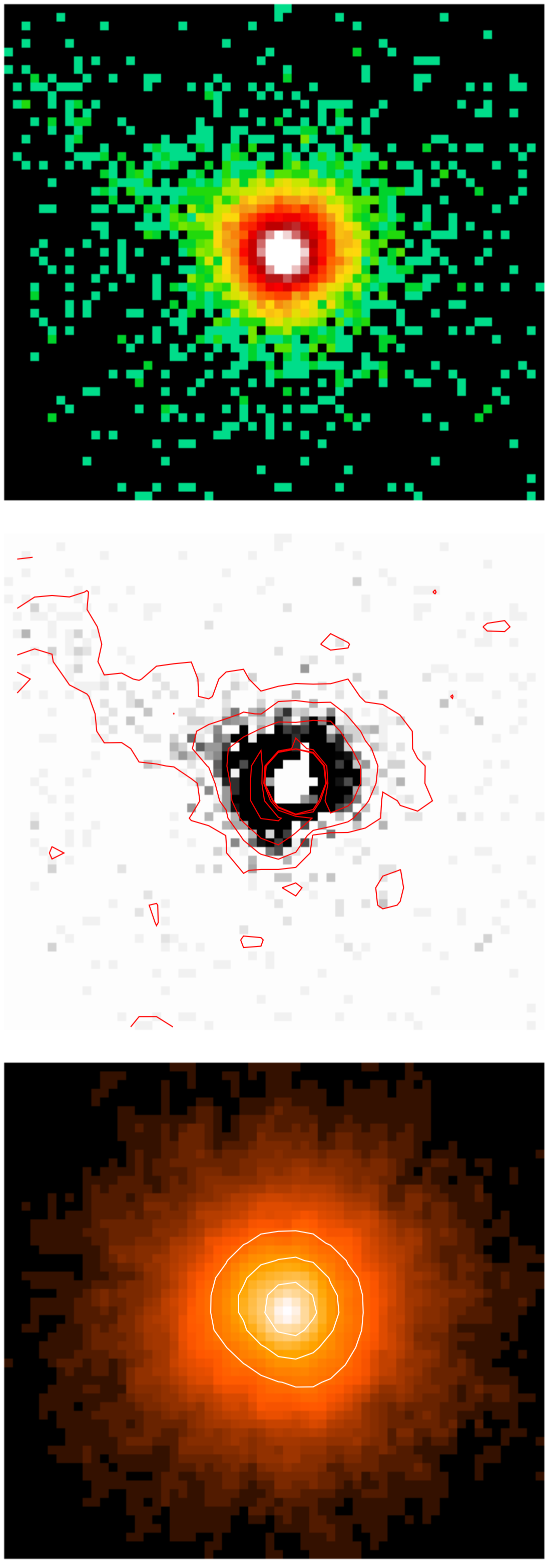


0.3–8 keV Radial Profile
100
10
1
Surf. Bright.(count/arcsec²)
5
10
15
Distance (arcsec)
SE Region
NW region
Total
PSF Model

Image Fitting in Sherpa

- Read data: binned image
- Read error image or use Sherpa statistics
- Display image with “image data”
- Filter the image interactively using ds9 or supply filter
- Define 2D models
- Use PSF as a model or convolution kernel
- Input exposure maps

```
Data 1: center_512_0.25pix.fits fits.
Total Size: 262144 bins (or pixels)
Dimensions: 2
Size: 512 x 512
Total counts (or values): 18130

Current Filters for dataset 1
ignore all
notice 1 filter box ( 255.5 , 257.25 , 59 , 55 , 0 )

Noticed Filter size: 3300 bins (pixels)

Total counts (or values) in filter: 16714

Optimization Method: Powell

Statistic: Cash

Current Models are:
source 1 = mypsf

Current Composite Models are:

Current Model Components are:
PSFfromFile[mypsf]
    Param   Type      Value        Min        Max                 Units
    -----   ----      -----        ---        ---                 -----
 1numCuts frozen          1          1          1
 2convTyp frozen          1          1          2
 3   file string: "psf_f1_onaxis_float_0.25pix.fits"
 4  xsize frozen         64          1       1024
 5  ysize frozen         64          1       1024
 6   xoff frozen          0       -512        512
 7   yoff frozen          0       -512        512
 8   xpos thawed   257.3017   225.5000   284.5000
 9   ypos thawed   257.4532   229.5000   283.5000
10   norm thawed 1.7091e+04          0      5e+04

sherpa> projection

Computed for projection.sigma = 1
        --------------------------------------------------------
        Parameter Name      Best-Fit Lower Bound     Upper Bound
        --------------------------------------------------------
         mypsf.xpos          257.302  -0.0145397      +0.0147531
         mypsf.ypos          257.453  -0.0158065      +0.0156077
         mypsf.norm          17091.2  -131.8          +132.496
```



Smoothed Images

- `CIAO tools: aconvolve, csmooth`

  aconvolve convolves an N-dimensional image with a kernel

  csmooth adaptive smoothing of a two-dimensional image (of integer counts) with a circular Gaussian or Tophat kernel

DIFFUSE EMISSION

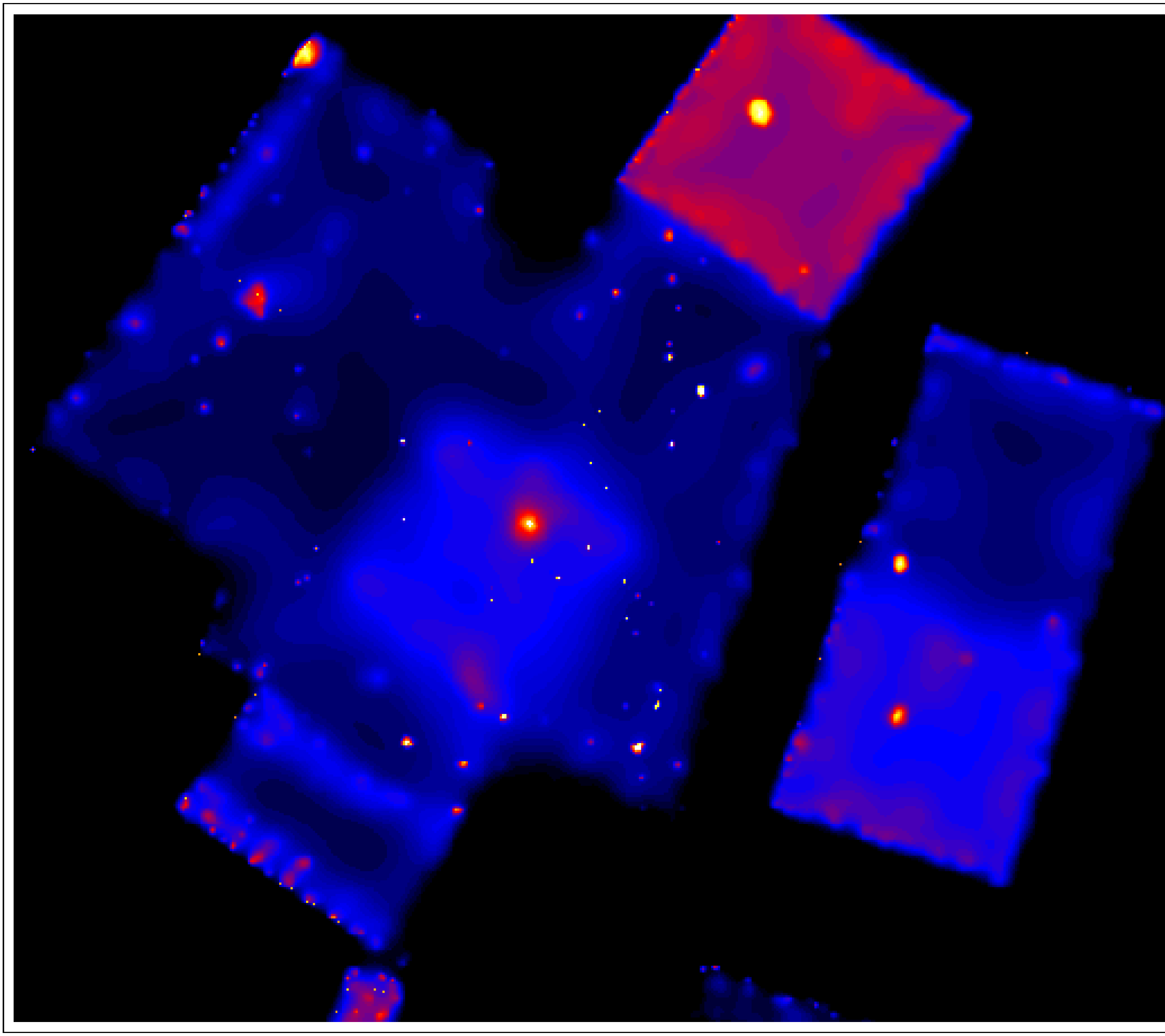


Figure 9: Image of the field obtained by the CIAO adaptive smoothing algorithm, CSMOOTH. Although much of the structure visible here is instrumental (we have not subtracted the background, which is different on the two types of chip), the extended emission around the nucleus, around NGC 604, and the elongated feature extended from the source south of the nucleus appear to be real.

Chandra X-ray Center

URL: http://cxc.harvard.edu/ciao2.2.1/threads/imag.html
Last modified: 2 October 2002

Imaging Threads for CIAO 2.2.1

- See the HRC Imaging Analysis Guide
- See the Extended Sources Analysis Guide
- Notes on Chandra Astrometric Accuracy

- Using SAOImage ds9
- Using CIAO Region Files
- How To Match the Binning of an Image
  See also: the `get_sky_limits.tar` script package
- Combining Data from Multiple Imaging Observations
  See also: the `combine_obsid` script
- Create A True Color Image
  See also: the `color_image` script
- Create an Image of Diffuse Emission
  See also: the `diff.scripts.tar` script package; the `merge_all.tar` script package
- Estimate Source Counts in an Image
- Obtain and Fit a Radial Profile
- Create a PSF

- Detect:
  - Detecting Sources in Imaging Observations − Overview
  - Detecting Sources in Imaging Observations − Using celldetect
  - Detecting Sources in Imaging Observations − Using vtpdetect
  - Detecting Sources in Imaging Observations − Using wavdetect
  - Using the Output of Detect Tools

- Exposure Maps:
  - Compute an HRC−I Exposure Map and Build Fluxed Image
    UPDATED (***1 Oct 2002***)
  - Use merge_all Script to Compute ACIS Exposure Maps and Fluxed Images
    See also: the `merge_all.tar` script package
  - Compute Single Chip ACIS Exposure Map and Fluxed Image Step−by−Step
  - Compute Multiple Chip ACIS Exposure Map and Fluxed Image Step−by−Step
  - Calculating Spectral Weights
    See also: the `spectrum.sl` S−Lang script

- Calculating Statistics of Images
  See also: the `sstats.sl` S−Lang script
- Reprojecting Coordinates of a Solar System Object

URL: http://cxc.harvard.edu/ciao2.2.1/threads/imag.html
Last modified: 2 October 2002