

CHANDRA
X-RAY OBSERVATORY



A Brief Intro to the Chandra Mission

Jonathan McDowell

CENTER FOR

ASTROPHYSICS

HARVARD & SMITHSONIAN

The Chandra X-ray Observatory



Launched 19.5 years ago 23 July 1999

A revolution in X-ray astronomy
and astronomy in general



What is Chandra?

NASA Chandra X-ray Observatory – operated by Chandra X-ray Center (CXC) at the Center for Astrophysics (Cambridge, MA) in collaboration with MIT

The greatest X-ray telescope ever built!

Orbits the Earth to be above the atmosphere

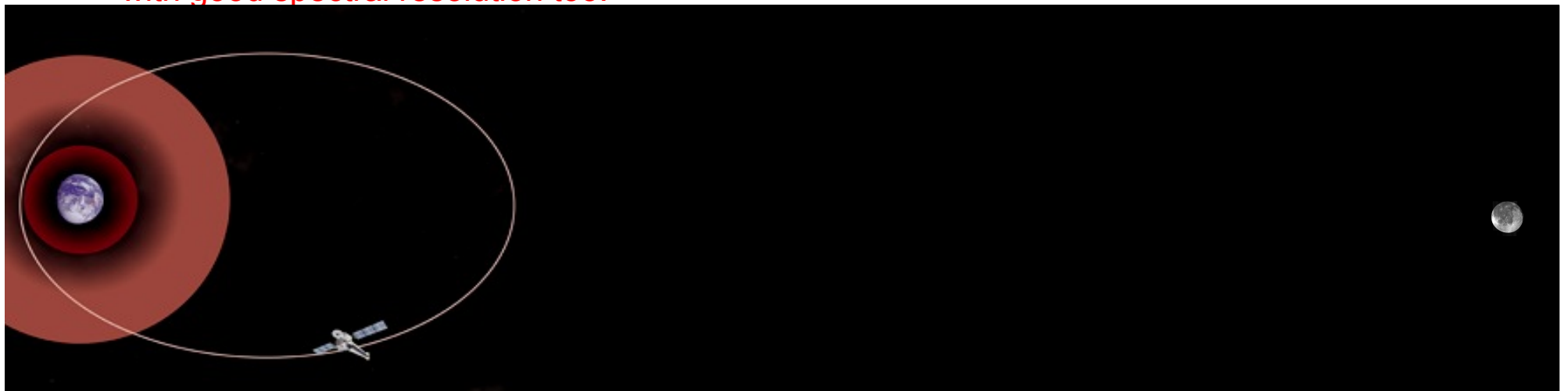
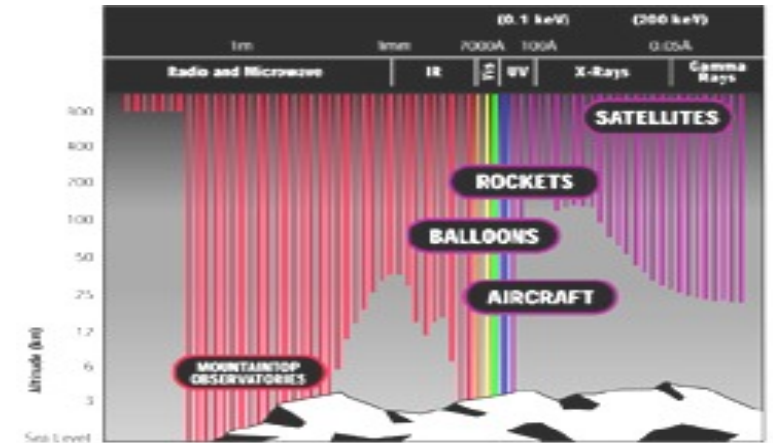
Goes 1/3 of the way to the Moon

every 64 hours (2 ½ days)

Observing only while outside radiation belts

Chandra takes superbly sharp images:

with good spectral resolution too!





X-ray Telescopes are different

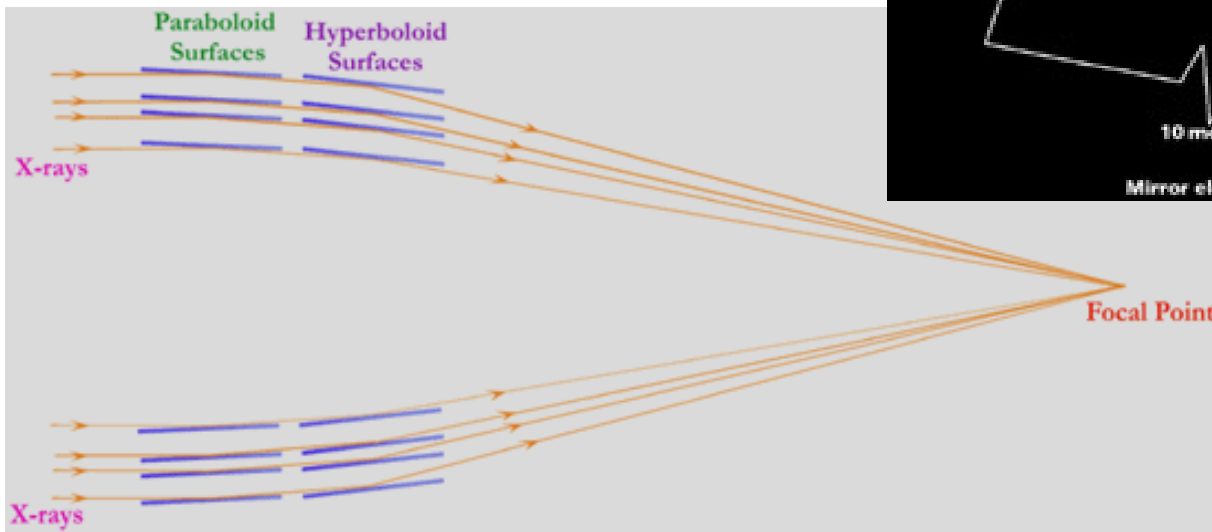
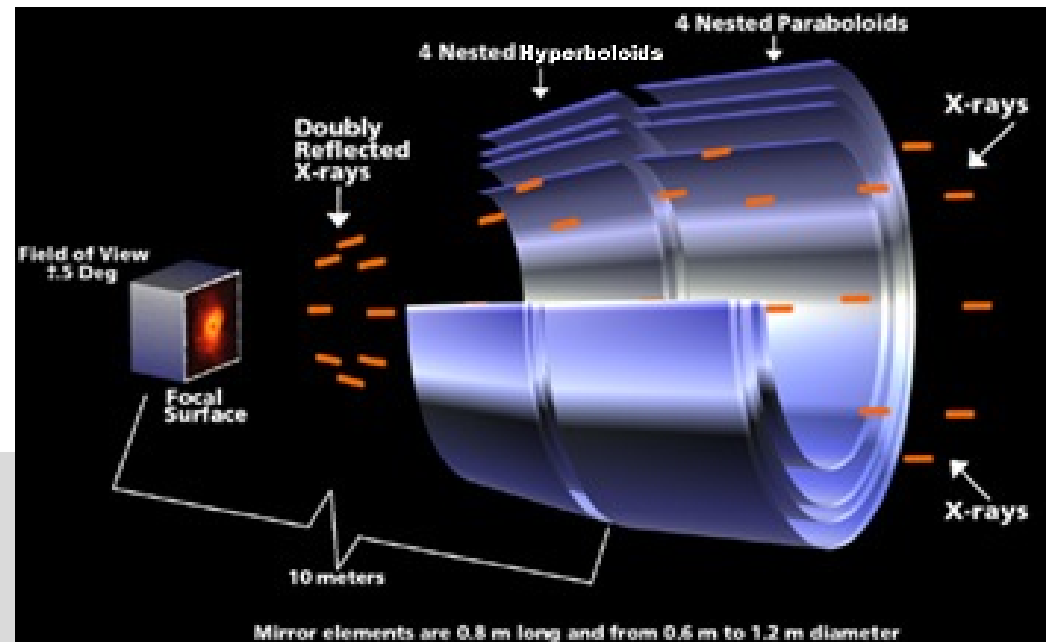
Chandra's mirrors are almost cylinders

X-rays don't reflect off a normal mirror – they get absorbed.

Only by striking a mirror at a glancing angle, about 1° , do X-rays reflect.

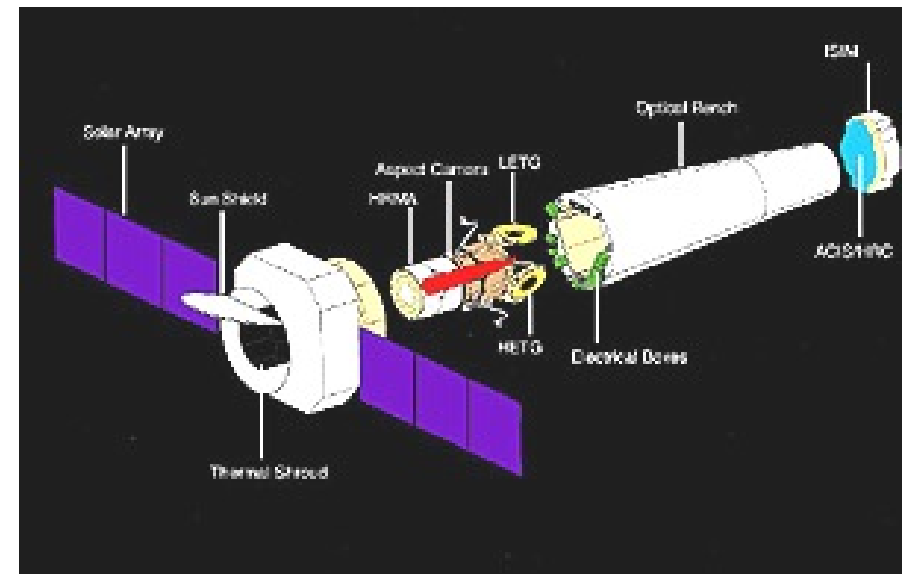
Focal length 10 m

4 mirror pairs -
perfectly aligned (not quite)



The Chandra spacecraft

10 meters (32 ½ ft) from mirror to detector, 1.2 meters (4ft) across mirror



...but focuses X-rays onto a spot only 25 microns across

Chandra Instruments and Gratings:

ACIS Advanced CCD Imaging Spectrometer

- Most Chandra observations use ACIS
- 10 x 1 Mpix chips, use any 4 (caveats)
- Sub arcsecond imaging on axis
- Each chip is 8 x 8 arcmin
- Spectrum in each pixel, 0.5 (ish) to 7 (ish) keV

HRC-I and HRC-S High Resolution Camera

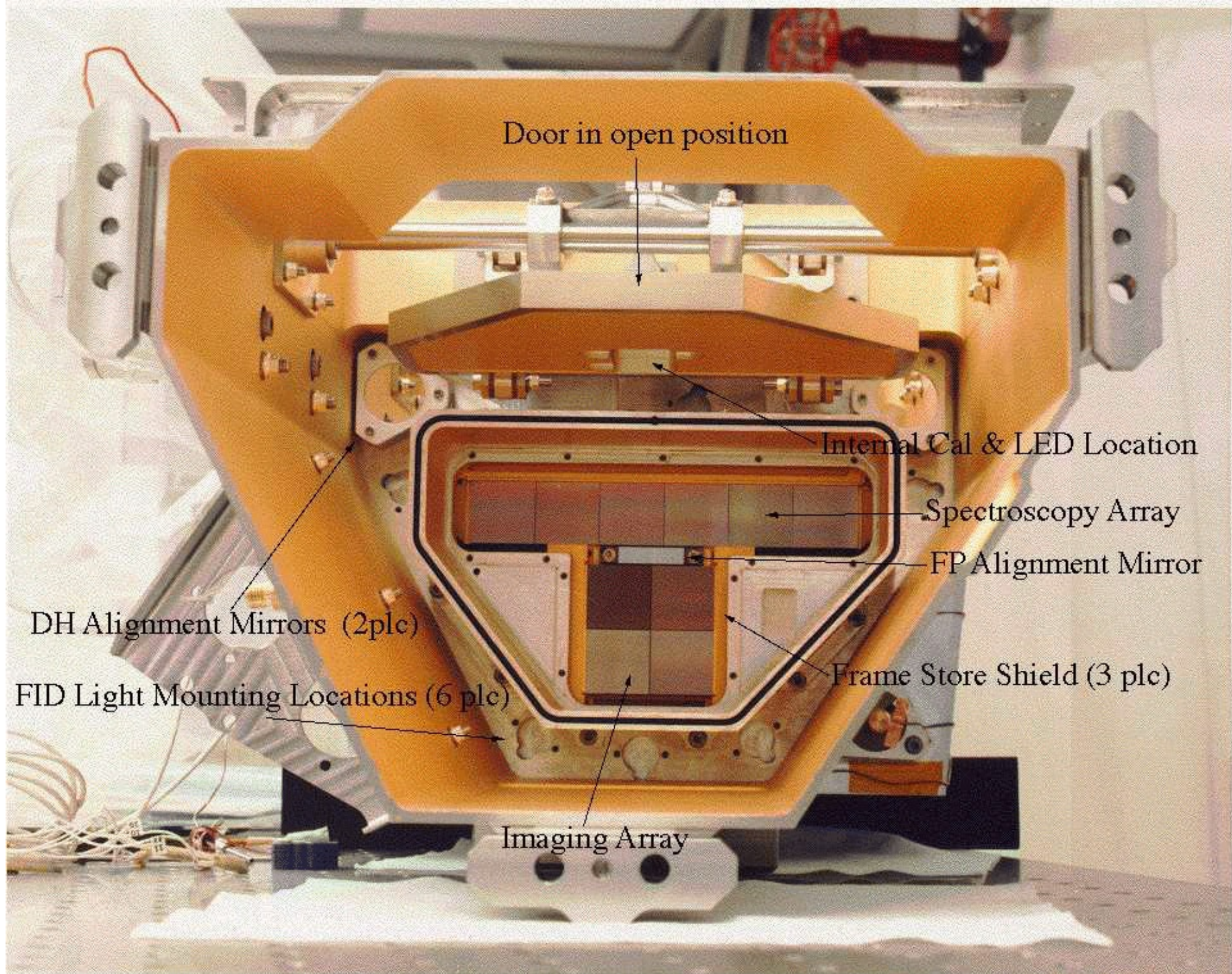
- Microchannel plate, highest spatial and timing res but no spectral res

HETG High Energy Transmission Grating

- Disperses spectrum onto ACIS
- Two gratings, HEG and MEG, at an angle to each other
- Order sorting possible via CCD energy discrimination

LETG Low Energy Transmission Grating

- Disperses spectrum onto HRC-S or ACIS
- Single dispersed spectrum
- Orders superimposed



Door in open position

Internal Cal & LED Location

Spectroscopy Array

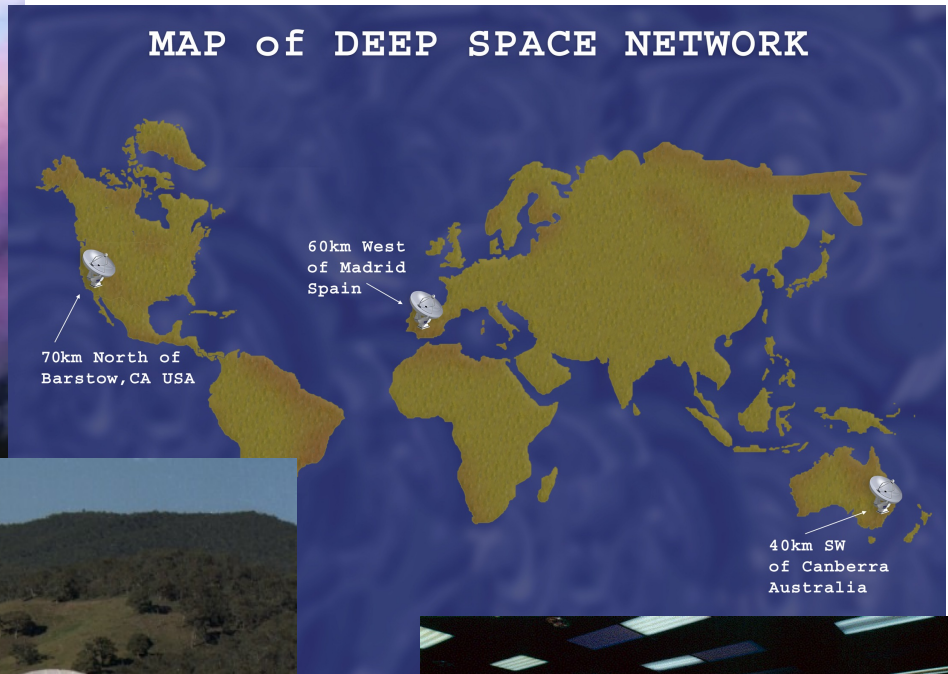
FP Alignment Mirror

Frame Store Shield (3 plc)

Imaging Array

DH Alignment Mirrors (2plc)

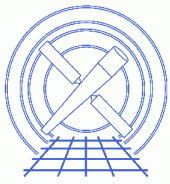
FID Light Mounting Locations (6 plc)



DSN control at Jet Propulsion Lab Pasadena, CA



Chandra science center Smithsonian Observatory, at Harvard (Cambridge, MA)



Chandra mission control Near MIT in Cambridge, MA

Main CIAO software release December each year
(typically)

Proposals due in March each year

Archive available all year round!! cxc.harvard.edu

CHANDRA
X-RAY OBSERVATORY



Introduction to X-ray Data Analysis

Jonathan McDowell

CENTER FOR

ASTROPHYSICS

HARVARD & SMITHSONIAN

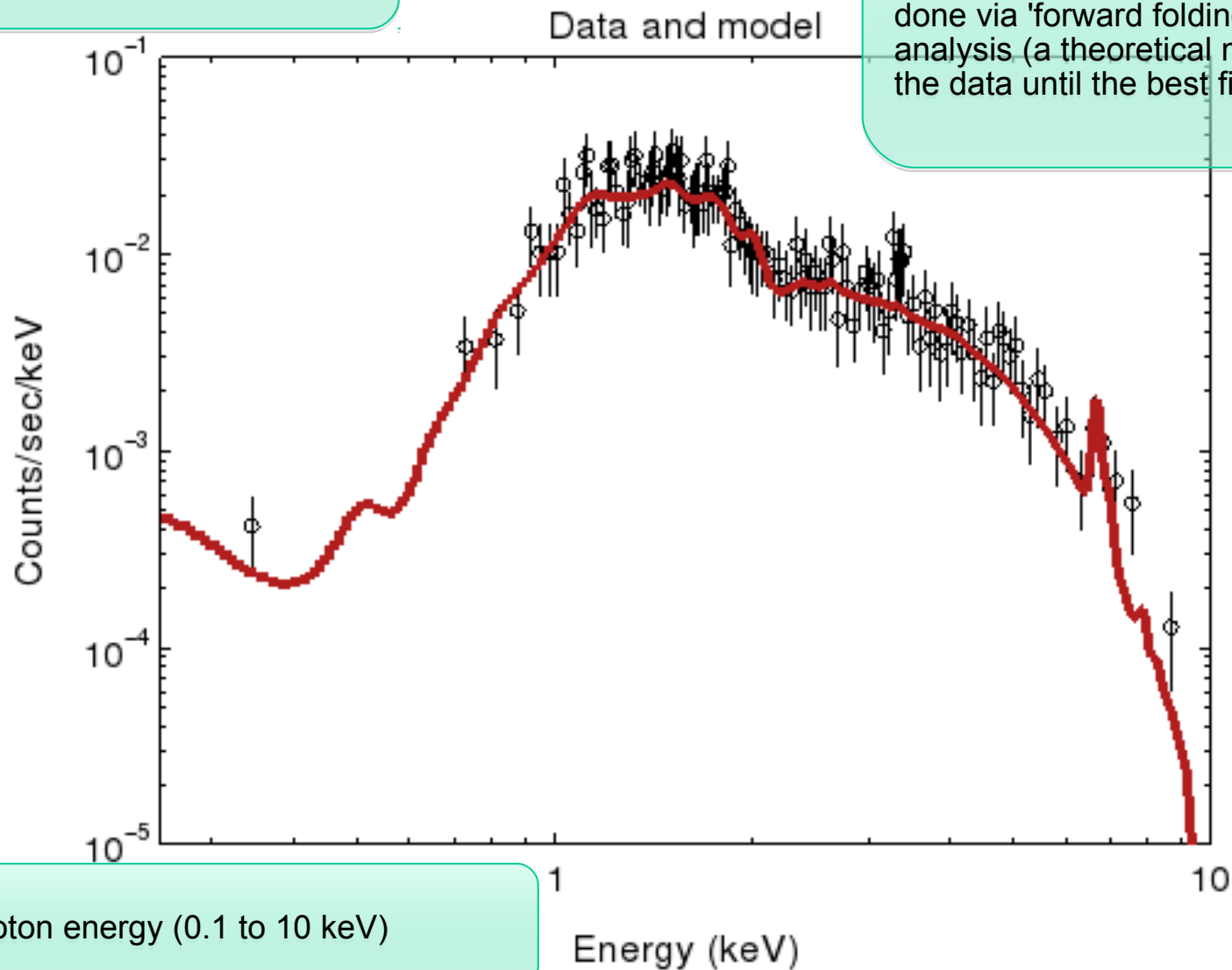
Introduction to X-ray Data Analysis

- X-ray astronomy is different
- **Problem 1:** Photon counting with small number statistics
- **Problem 2:** Spectral line spread function is often broad and messy - forced to forward-folding approach
- **Problem 3:** Bands are very broad, so energy (wavelength) dependence more obvious (e.g. in PSF)
- **Problem 4:** Different optics - PSF degrades rapidly off axis
- **Problem 5:** The telescope is not pointing steadily like, say, HST - it's moving back and forth across the source.
- But:
- **Advantage:** We have more information on each photon (position, energy, arrival time)

Complexities in X-Ray and Chandra Data Analysis

In Poisson statistics regime because of the small number of photons

Imaging data has limited energy resolution and modeling can only be done via 'forward folding' spectral analysis (a theoretical model is fitted to the data until the best fit is found)

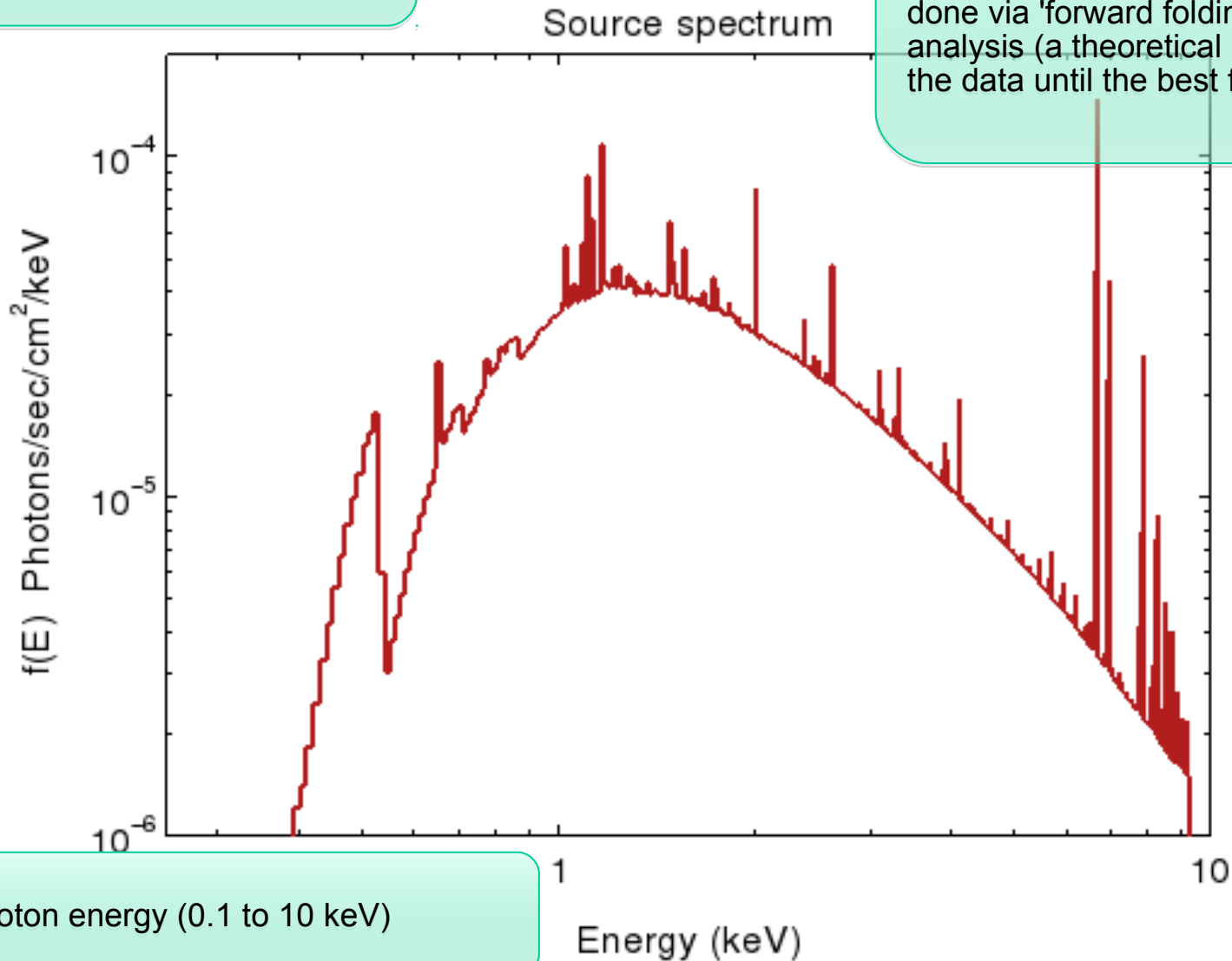


2 decades of photon energy (0.1 to 10 keV)

Complexities in X-Ray and Chandra Data Analysis

In Poisson statistic regime because of the small number of photons

Imaging data has limited energy resolution and modeling can only be done via 'forward folding' spectral analysis (a theoretical model is fitted to the data until the best fit is found)



2 decades of photon energy (0.1 to 10 keV)

Complexities in X-Ray and Chandra Data Analysis

Every aspect of the observation varies with

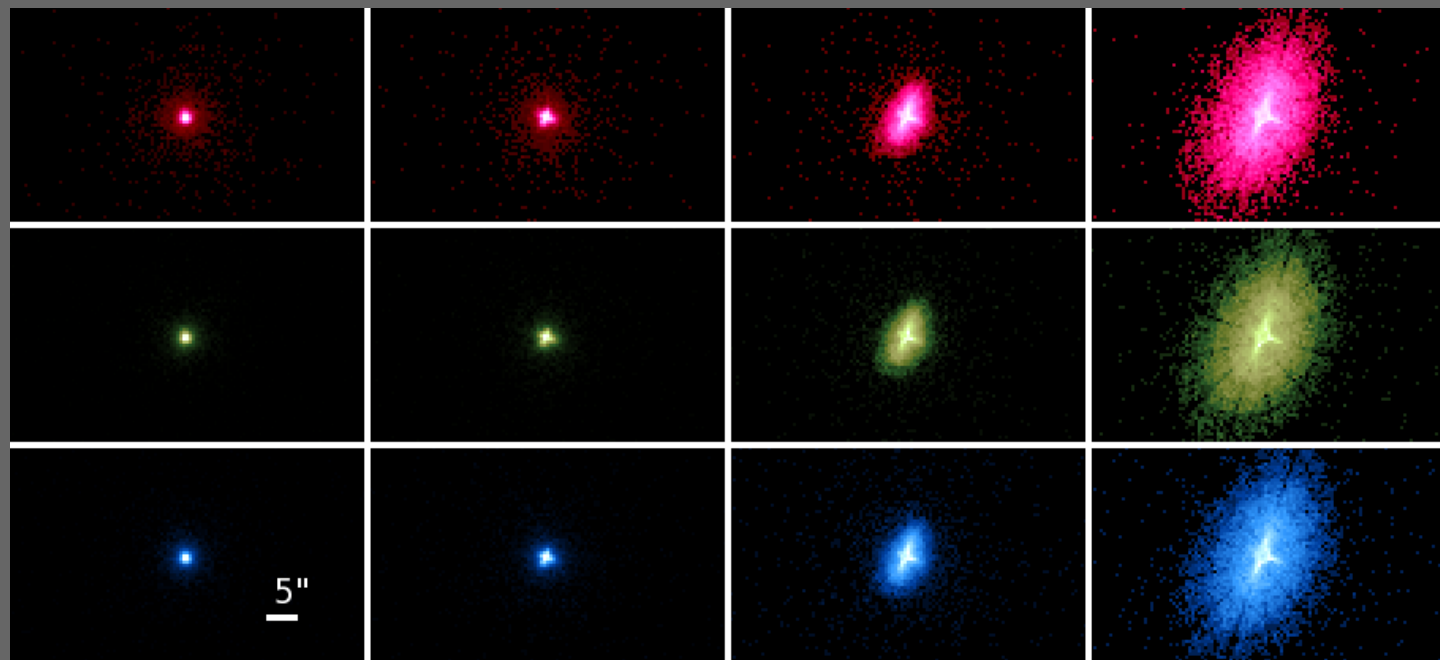
- energy
- position
- time

(e.g. image sharpness, sensitivity, instrumental energy scale)

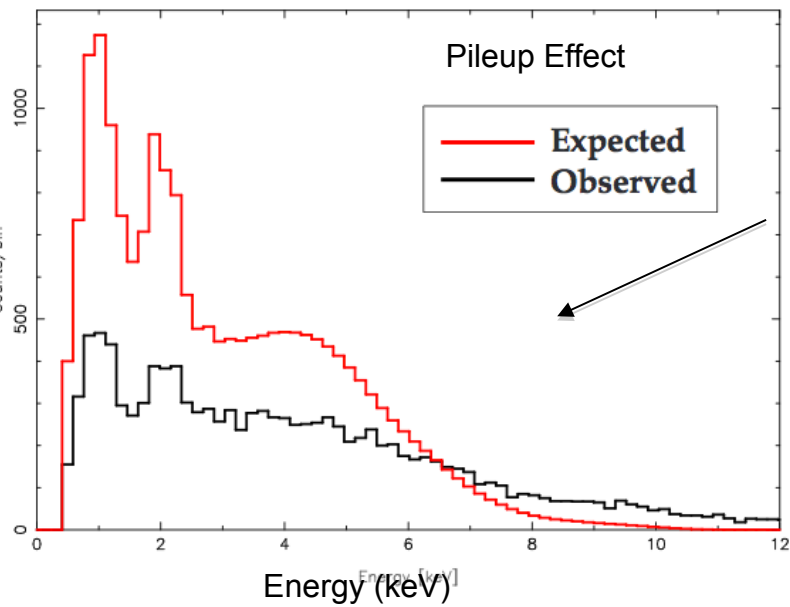


Energy

The Chandra PSF



Q0836+7104 predicted vs. observed



Pileup Effect

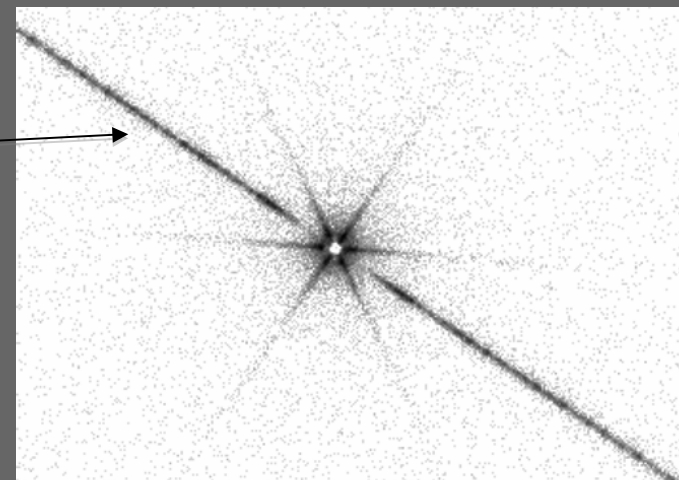
- Expected
- Observed

Specific instrumental effects eg.

readout streak

pileup - two or more photons detected as single event

Off-axis angle



Basics of CIAO

- Data files are in FITS format (usually binary tables, not just images)
- CIAO can also operate on ASCII files in many cases
- All (well, almost all) CIAO tools that want an input file can accept a CIAO Data Model “virtual file”
 - e.g instead of `evt.fits`
 - take `“evt.fits[energy=300:1000,sky=circle(4096,4096,20)]”`

Each file (dataset) is made up of sections called 'blocks' (HDUs for FITS fans)
Blocks can be tables or images

Key tools:

```
dmcopy infile outfile  
dmlist infile opt=blocks,cols,data
```

```
ahelp dmlist → help for tool dmlist  
plist dmlist → list parameters for dmlist
```

Key applications:

```
Sherpa - fitting  
ChIPS – plotting (being superseded by matplotlib)  
ds9 – imaging and analysis
```

The Event File

- In optical astronomy, the primary data set is an image. In radio interferometry, it's a visibility array.
- In X-ray astronomy, the primary data set is an event list - a table of (putative) photons
 - Our software makes it easy to generate an image from the event list, so it's easy to forget that's what you have. But making the image loses information.
 - First cut way of thinking about the event list: it's a 4-dimensional array of x, y, time, energy. But most pixels are empty (we don't have many photons!) so it's more compact to just list the non-empty ones.
 - Complication: we actually have many more parameters for each photon, not just 4.

Inside the event list

```
jupiter> dmlist acisf03041_001N001_evt3.fits cols
```

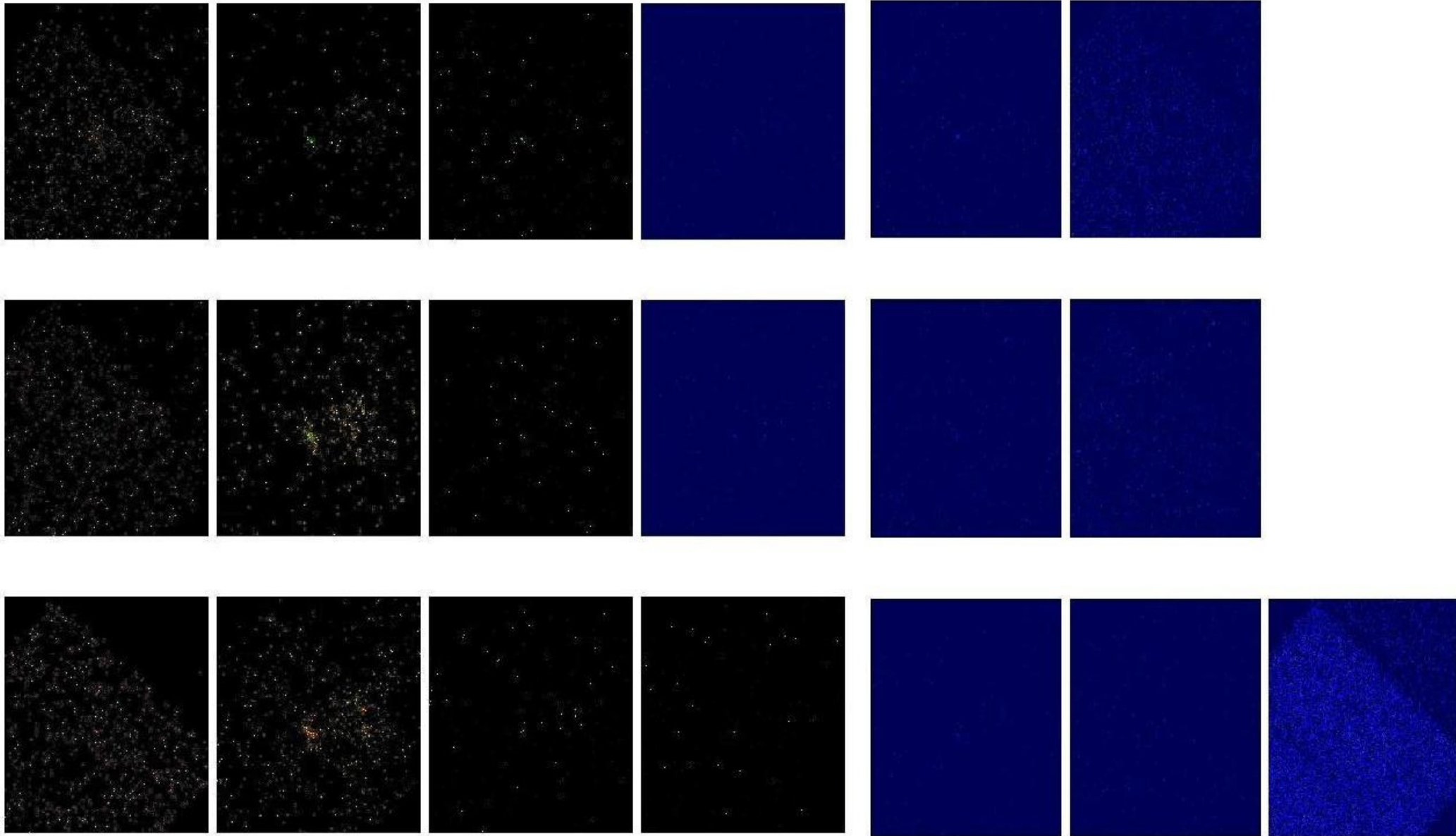
Columns for Table Block EVENTS

ColNo	Name	Unit	Type	Range	
1	time	s	Real8	154361559.6127299964:154436827.4158599973	S/C TT corresponding to mid-exposure
2	ccd_id		Int2	0:9	CCD reporting event
3	node_id		Int2	0:3	CCD serial readout amplifier node
4	expno		Int4	0:2147483647	Exposure number of CCD frame containing event
5	chip(chipx,chipy)	pixel	Int2	1:1024	Chip coords
6	tdet(tdety,tdey)	pixel	Int2	1:8192	ACIS tiled detector coordinates
7	det(detx,dety)	pixel	Real4	0.50: 8192.50	ACIS detector coordinates
8	sky(x,y)	pixel	Real4	0.50: 8192.50	sky coordinates
9	pha	adu	Int4	0:36855	total pulse height of event
10	pha_ro	adu	Int4	0:36855	total read-out pulse height of event
11	energy	eV	Real4	0: 1000000.0	nominal energy of event (eV)
12	pi	chan	Int4	1:1024	pulse invariant energy of event
13	fltgrade		Int2	0:255	event grade, flight system
14	grade		Int2	0:7	binned event grade
15	status[4]		Bit(4)		event status bits

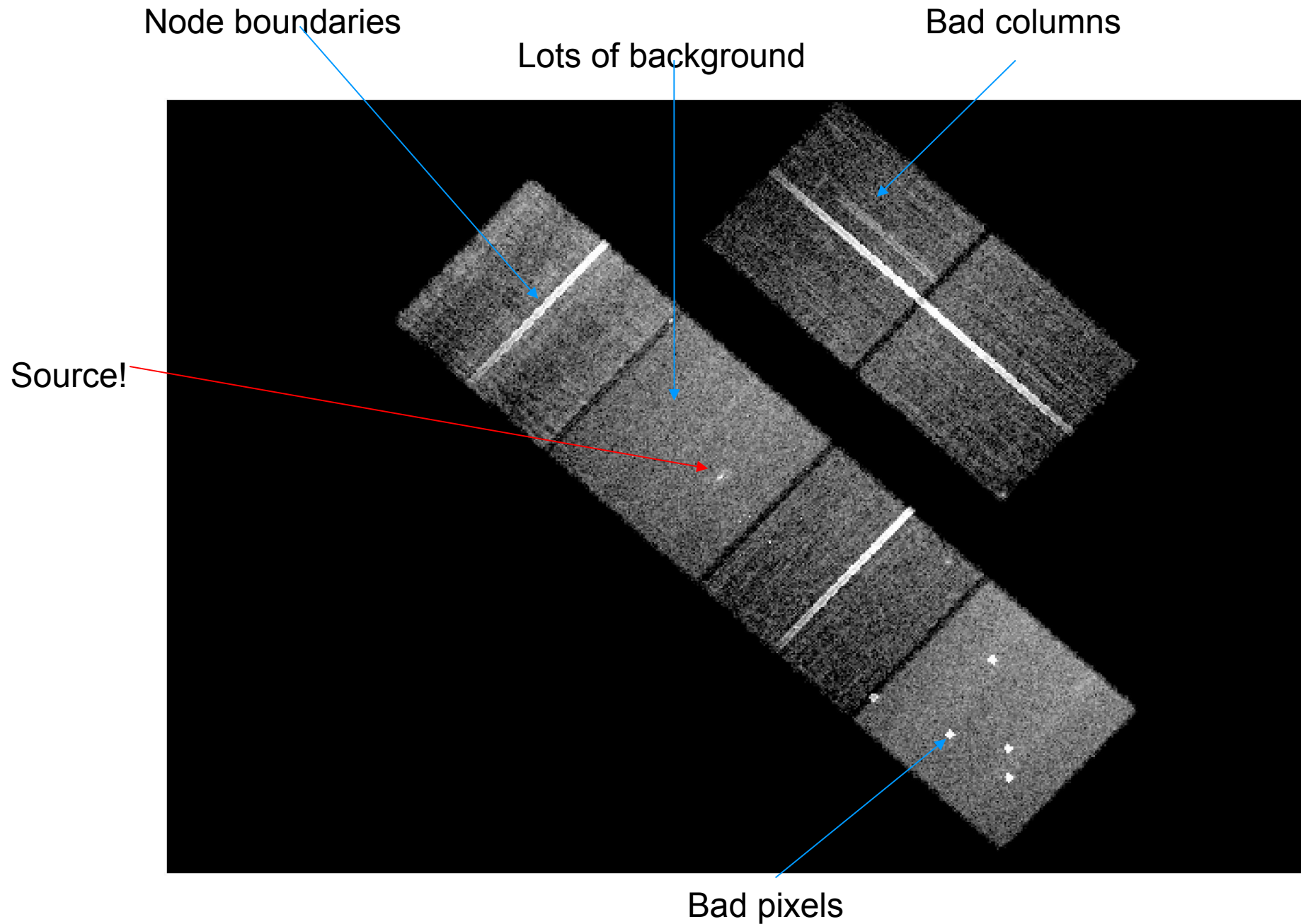
```
jupiter> dmlist acisf03041_001N001_evt3.fits["cols -status]" data,raw,clean rows=1:20
```

#	time	ccd_id	node_id	expno	chip(chipx,chipy)	tdet(tdety,tdey)	det(detx,dety)	sky(x,y)	pha	pha_ro	energy	pi	fltgrade	grade
154362662,7665936351	0	1	107	369 513	3574 4763	3540,2504882812	1556,8157958984	1473,2664794922	4129,1977539062	3688	3680	15358,6318359375	1024	16 4
154362662,7665936351	0	2	107	562 589	3650 4570	3615,2900390625	1748,9260253906	1676,4797363281	4093,9599609375	3977	3750	15254,2246093750	1024	64 2
154362662,7665936351	0	0	107	247 876	3937 4885	3902,1516113281	1435,5321044922	1426,5589599609	3750,3830566406	3765	3514	14473,1611328125	992	0 0
154362662,8076336384	7	0	107	189 301	4106 2003	4069,4038085938	4313,6518554688	4280,3339843750	4160,218750	3568	3503	15899,1279296875	1024	11 6
154362662,8076336384	7	1	107	264 388	4181 2090	4144,4223632812	4225,7763671875	4209,175781250	4069,1887207031	128	109	632,9125366211	44	72 6
154362662,8076336384	7	2	107	555 410	4472 2112	4435,0400390625	4204,0610351562	4245,8291015625	3780,0749511719	1717	1702	7969,5327148438	546	8 3
154362662,8076336384	7	2	107	676 441	4593 2143	4556,0268554688	4172,6386718750	4239,1547851562	3655,2526855469	1908	1853	8829,906250	605	16 4
154362662,8076336384	7	1	107	483 465	4400 2167	4363,3208007812	4149,5029296875	4178,0693359375	3839,4790039062	1011	994	4767,4877929688	327	2 2
154362662,8076336384	7	3	107	881 613	4798 2315	4760,8256835938	4001,3994140625	4112,1772460938	3420,4289550781	1348	1310	6024,1176757812	413	0 0
154362662,8076336384	7	2	107	690 834	4607 2536	4569,7802734375	3780,8713378906	3857,9919433594	3563,6794433594	2011	1942	9294,6806640625	637	72 6
154362662,8076336384	7	1	107	348 925	4265 2627	4228,7290039062	3689,4941406250	3700,4628906250	3879,6706542969	1722	1664	8069,8139648438	553	2 2
154362662,8076336384	7	1	107	502 954	4419 2656	4381,6430664062	3660,5825195312	3702,6135253906	3724,0622558594	3011	2957	14100,3310546875	966	208 6
154362662,8486736417	6	3	107	803 548	3678 2250	3639,6469726562	4066,6665039062	3952,636718750	4532,1162109375	2258	2095	8640,0263671875	592	2 2
154362662,8897136450	3	0	107	40 717	4415 3101	4376,0224609375	3215,3222666250	3265,1691894531	3640,8110351562	3146	2950	12326,56250	845	0 0
154362662,8897136450	3	0	107	78 901	4231 3139	4192,5224609375	3177,2634277344	3191,2951660156	3813,0412597656	2252	2077	8826,3281250	605	64 2
154362662,9307536483	2	0	107	208 240	3301 3878	3266,0554199219	2440,0964355469	2284,2392578125	4573,9658203125	3451	3423	13226,7988281250	906	16 4
154362662,9307536483	2	2	107	517 719	3780 3569	3744,9583984375	2748,1215820312	2681,5275878906	4166,1748046875	3816	3544	14376,746093750	985	64 2
154362662,9717936218	1	0	107	101 567	4565 4208	4527,886718750	2109,5788574219	2211,8908691406	3271,5732421875	1644	1543	6498,9438476562	446	64 2
154362662,9717936218	1	3	107	996 952	4180 5103	4141,839843750	1216,6264648438	1259,9039306641	3471,8693847656	2406	2170	9509,4218750	652	16 4
154362666,0075938106	0	2	108	683 156	3217 4449	3183,1049804688	1869,4627685547	1709,3774414062	4541,6933593750	3466	3429	13290,0634765625	911	0 0

Energy slices through an event list, 0.1 - 10 keV



Level 1 Event List - Calibrated but Dirty



Level 2 event list - cleaned and filtered

Energy filter 300-7000 eV removes background but not signal

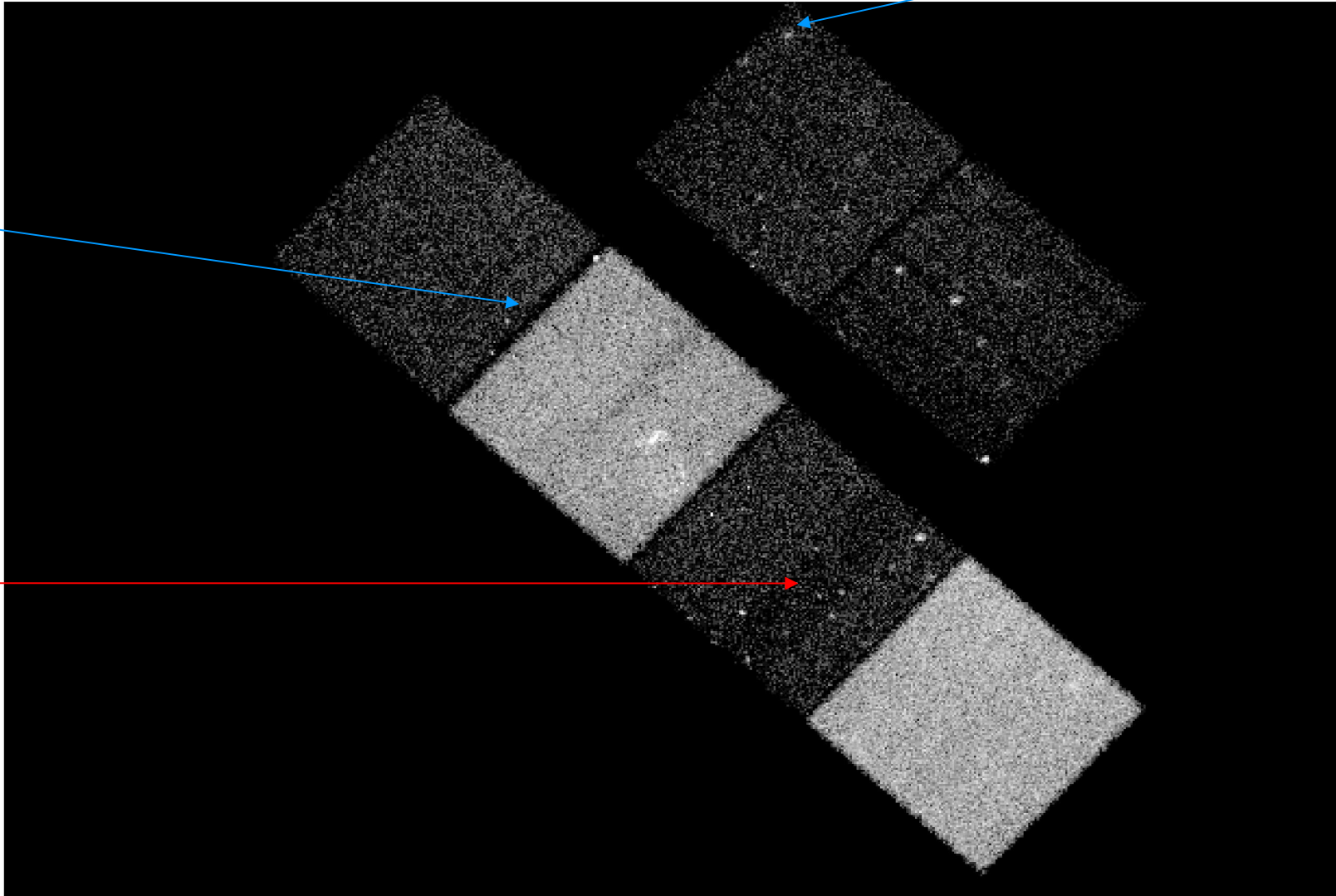
Grade filter removes cosmic ray events etc

Good time filter removes times of high background, poor data quality

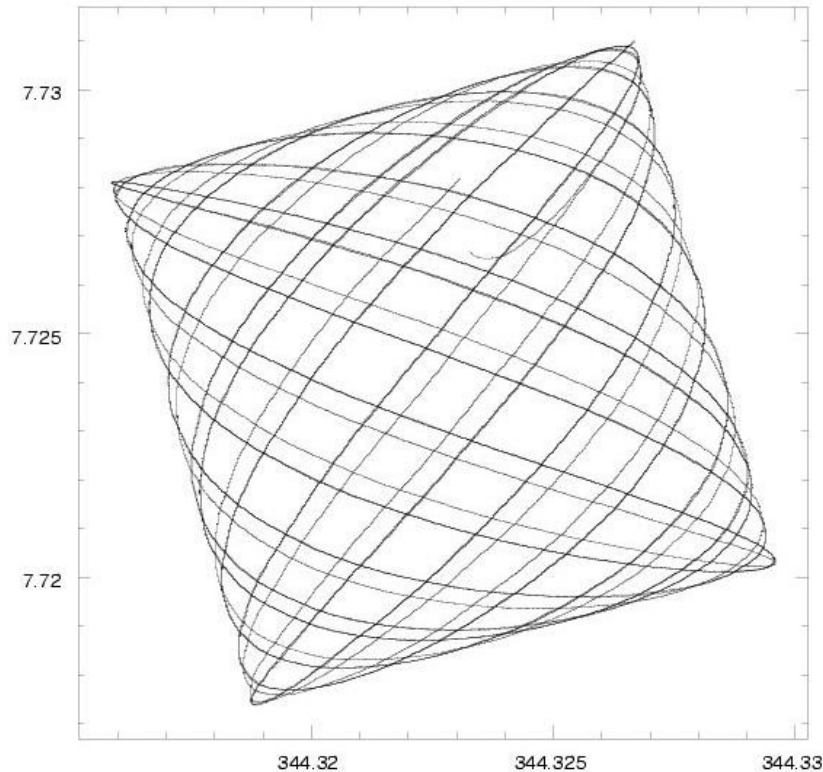
Sources fuzzy far
off axis (PSF big)

Beware
chip gaps!

More
sources!



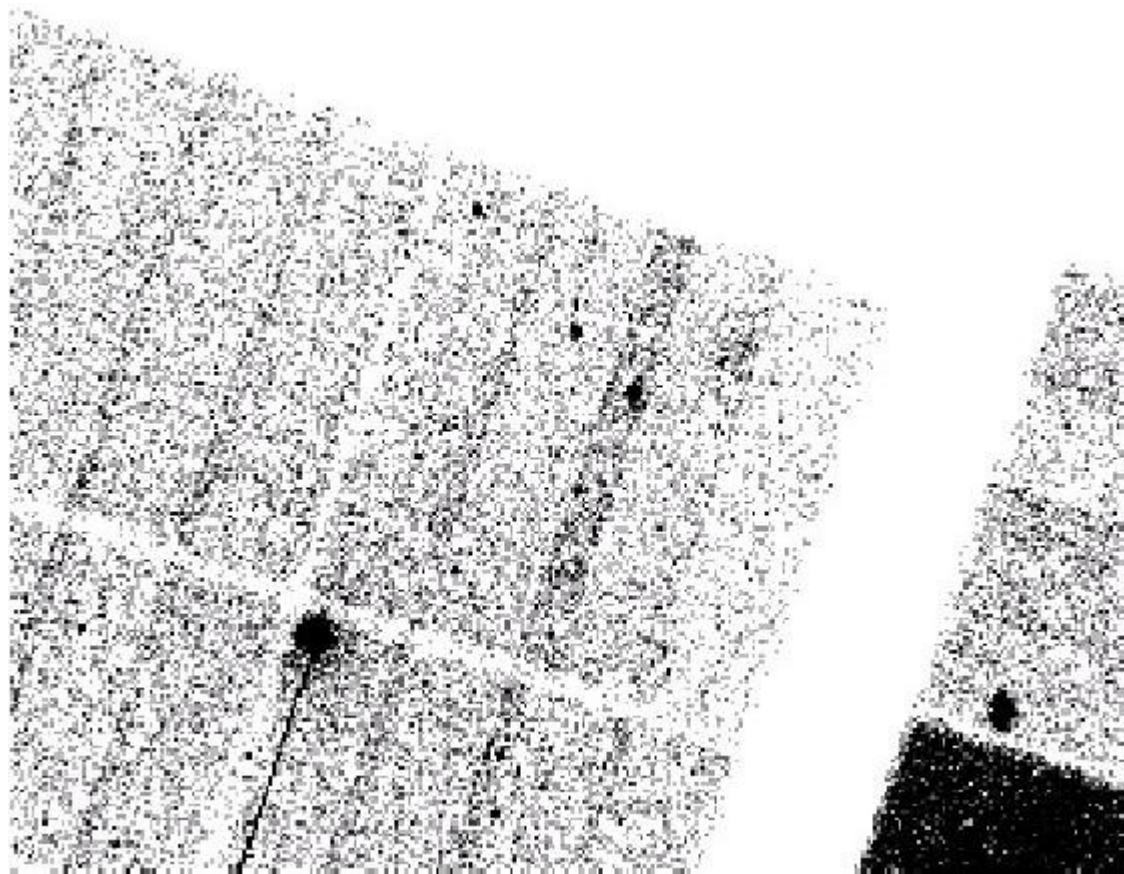
The aspect solution



During an observation, Chandra's optical axis describes this 'dither pattern' on the sky, (Problem 5), smearing the image of a point source. The RA, Dec, roll angle of the telescope versus time is called the 'aspect solution'; the asol1.fits file provides this for each observation.

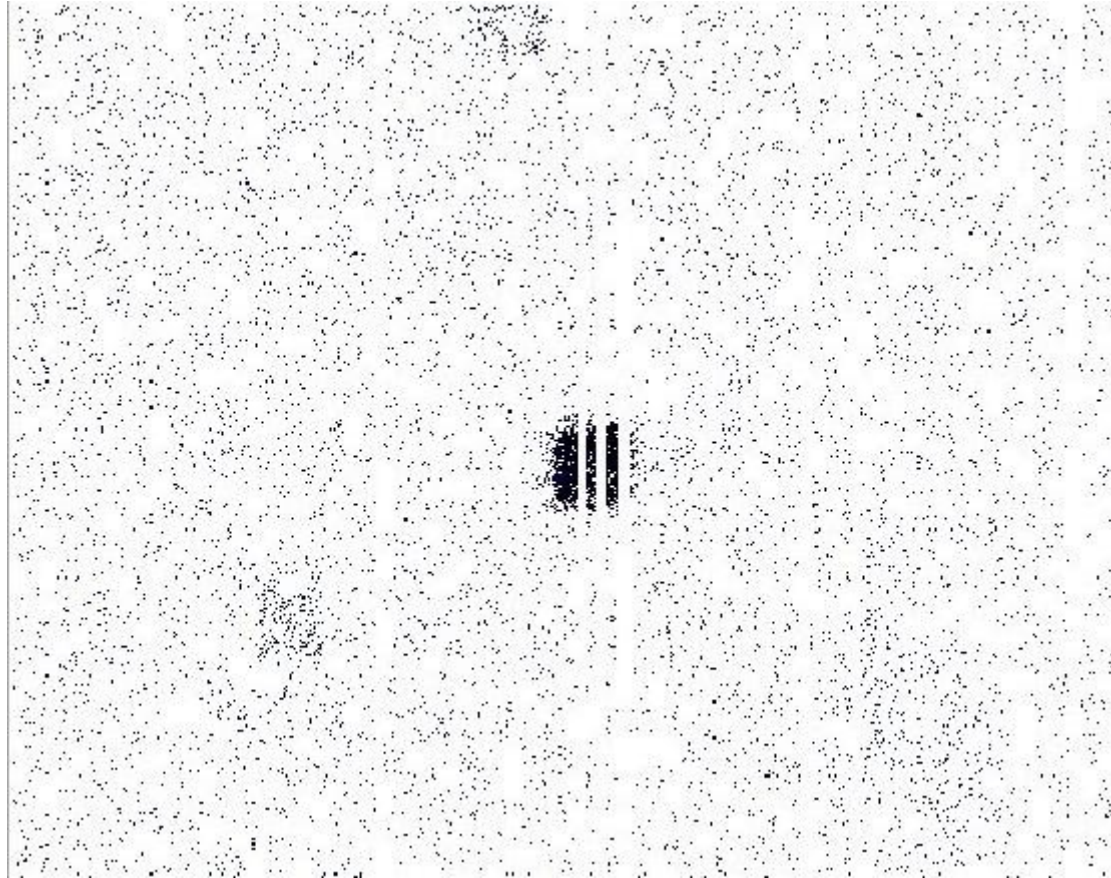
We record the motion of the guide stars in the star tracker so that we can calculate RA and Dec for EACH PHOTON and so reconstruct the image.

Chandra aspect-corrected data



This is what you get after calibration but before cleaning the data. Note the sharp point sources near the center.

Chandra raw (chip) data



In instrument space, the photons are spread out over 20 arcsec and have bad columns going through them - so be careful of the effective exposure time. If you didn't dither, you could lose the source entirely if it landed on a bad pixel

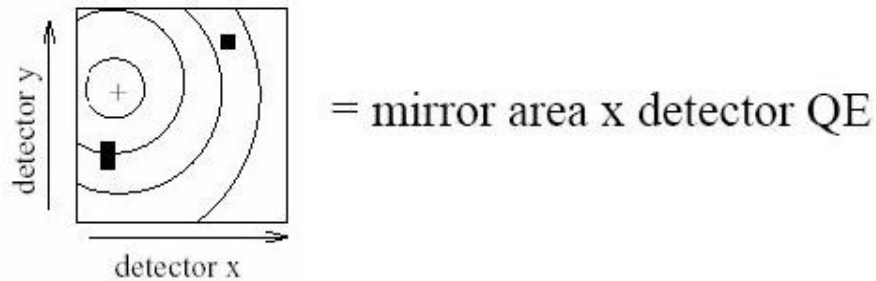
Spatial Response: EXPOSURE MAP

The *Exposure Map*, $E(\Delta h, \lambda, \hat{p})$ retains spatial information at the expense of spectral. It has units of $[\text{cm}^2 \text{ counts photons}^{-1}]$.

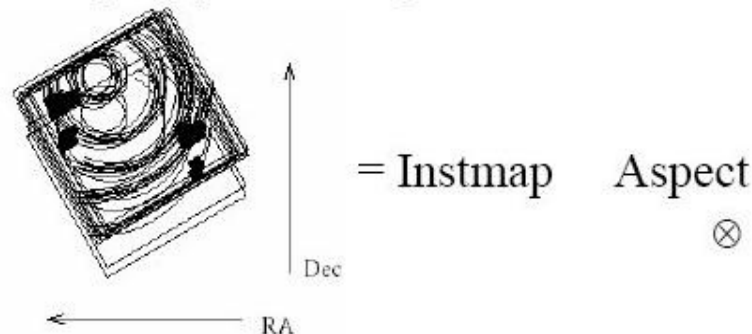
$$\int d\lambda S(\lambda, \hat{p}) \approx \frac{C(\Delta h, \hat{p})}{E(\Delta h, \lambda, \hat{p})}$$

C is the observed counts per spatial bin in a pulse-height bin. S is the source flux, with units of $[\text{phot cm}^{-2}\text{s}^{-1}\text{\AA}^{-1}]$.

Instrument Map – efficiency calibration information, band integrated. (create with `mkinstmap`)



Exposure Map – applies telescope aspect history and coordinate transformations (= area x time). (create with `mkexpmap`).



Event analysis or binned analysis?

- Don't make an image too quickly. If you can get an answer directly from the event list, that's better - binning the data loses information, and collapsing the axes loses information.
- Spatial analysis: make an image (using `dmcopy`)
 - lose energy and time information
- Spectral analysis: make a 'PHA file' using `dmextract` (or a grating spectrum using `tgextract`)
 - lose spatial and time information
- Temporal analysis: make a light curve using `dmextract`

The fundamental equation of astronomy

$$N(E) = A(E)F(E)\Delta T$$

Our instrument makes a spectrophotometric measurement; the sensitivity (“effective area”) $A(E)$ tells us how to convert from flux to instrumental counts for a given exposure time ΔT

But, a real instrument doesn't measure the true energy, it measures instrumental energy E' . The line spread function (“response matrix” in X-rays) $R(E,E')$ describes how a monochromatic input spectrum is broadened by the instrument (**Problem 2**)

Let us further assume that the instrumental energy E' is measured in discrete channels (bins) E'_i . Then

$$N(E'_i) = \int A(E)R(E, E'_i)F(E)dE\Delta T$$

Of course, you may not be measuring all of the light from the source. Even if it's a point source, there may be an aperture correction. We need the PSF $P(x-x',y-y')$ and the spatial dependence of the QE, $q(x,y)$. Then at a given instrument position x',y'

$$N(E'_i, x'_i, y'_i) = \int \int A(E)R(E, E'_i)F(E, x, y)P(x-x'_i, y-y'_i)q(E, x'_i, y'_i)dEdxdy\Delta T$$

The source may also be variable in time - we'll ignore this for the purposes of this talk. The detector sensitivity is time-variable on long timescales, but for a single observation you just have to worry about times when the data is filtered - the Good Time Intervals (GTIs)

$$N(E'_i, x'_i, y'_i) = \int \int \int A(E)R(E, E'_i)F(E, x, y, t)P(x-x'_i, y-y'_i)q(E, x'_i, y'_i)dEdxdydt$$

Pulse height

When you plot an optical spectrum, the wavelength (or energy) axis is really an instrumental quantity. A spectral line is broadened by instrumental effects, so the energies plotted are not the true energies of the photon. However, the instrument is calibrated (i.e. the definition of instrumental energy is rescaled) such that the peak of a line is at the correct energy.

In X-ray astronomy, instead of using the instrumental energy E' , we work with the energy bin number. For historical reasons to do with long-forgotten instruments, this bin number is known as the PI channel (for 'pulse invariant' channel) - we'll denote it by P . So, for fixed energy bin widths dE ,

$$E' = P dE = [\text{on average}] E$$

The instrument actually measures a raw energy bin number p , called the PHA channel, or 'pulse height analyser channel'. The scaling of the instrumental energy to real energy depends on position and time:

$$E'(\text{raw}) = p dE = g(x,y,t)P dE$$

This function g (the gain) is usually assumed to obey

$$g(x,y,t) = g_{\text{spatial}}(x,y) g_{\text{t}}(t)$$

and we provide calibrations of both the spatial gain and the temporal gain.

Spectra in Poissonland

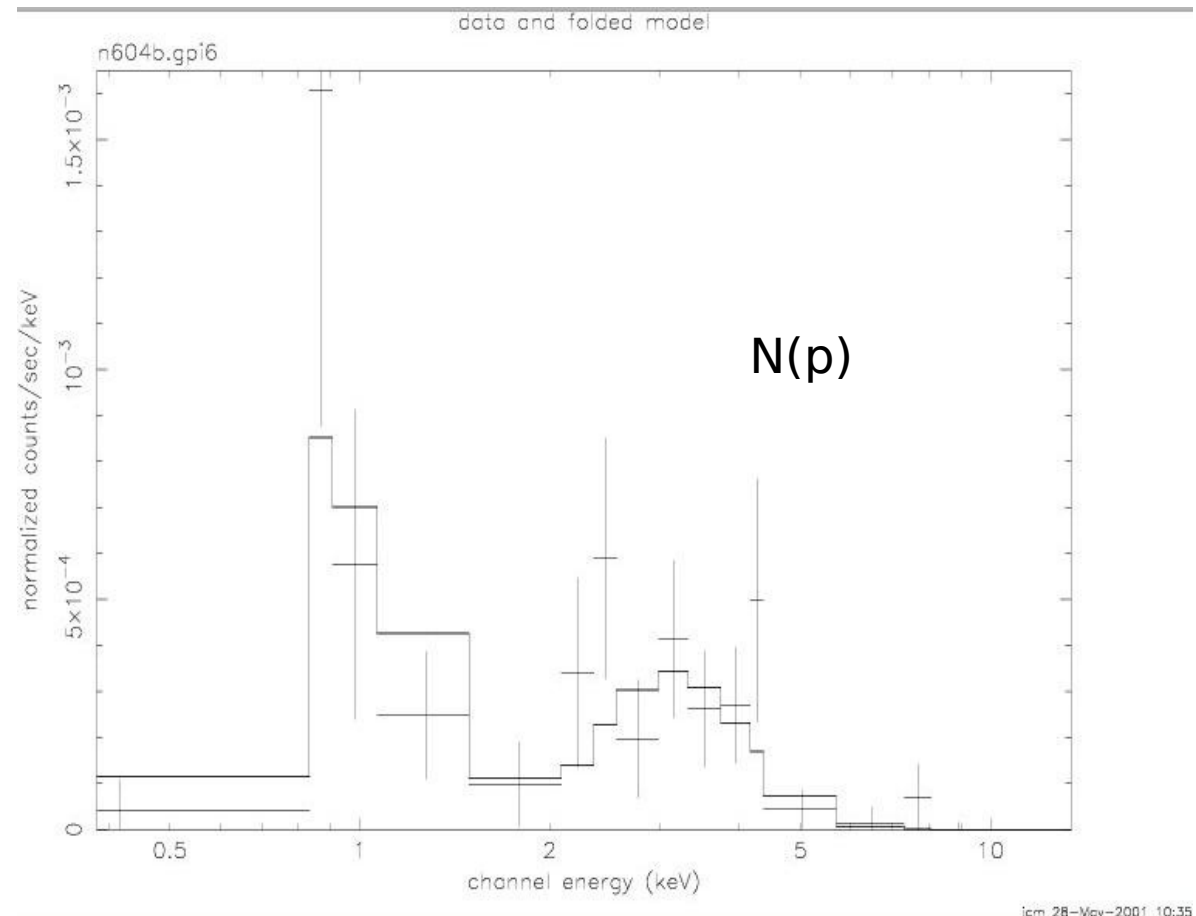
$$N(p) = \int R(E, p) A(E) F(E) dE$$

We pick a parameterized $F(E)$ such as warm absorber models, lines, thermal plasma codes. Which $F(E)$? You must pick one based on expected physics, but match number of free parameters with quality of data.

With less than 100 counts, we usually just use count ratios (X-ray colors) for spectral analysis.

Does one model fit significantly better than another? Be careful that two physically different models may look quite similar in $F(E)$ space.

Incompletely calibrated instrumental features may show up in residuals, limiting factor in high S/N spectra - these features may include edges. Beware apparent science in regions where $A(E)$ is changing rapidly.

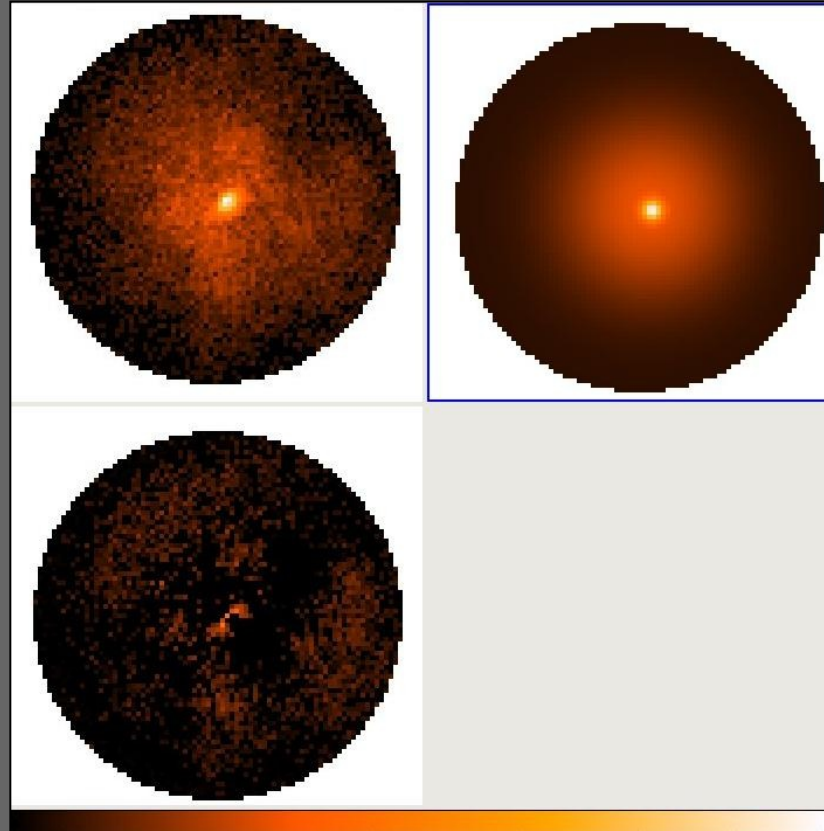
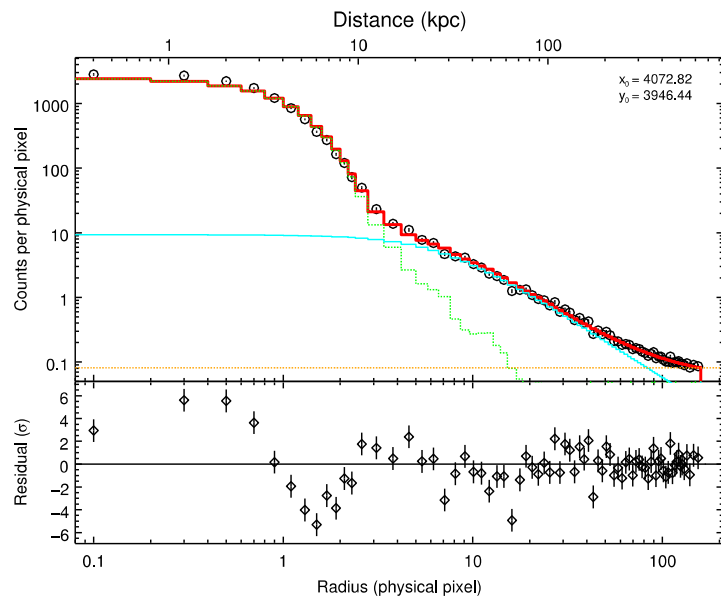


Sherpa: Modeling and Fitting in Python



Sherpa

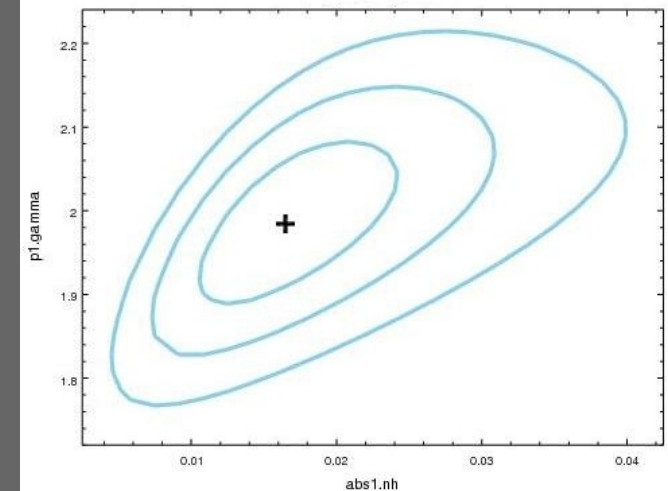
Modeling and fitting for 1-D and 2-D datasets **in any waveband** including:
spectra, images, surface brightness profiles, light curves, general ASCII data.



Coded in a Python environment – familiar to the new generation of astronomers and used in other missions

Model Poisson and Gaussian data

Calculate confidence levels on the best-fit model parameters

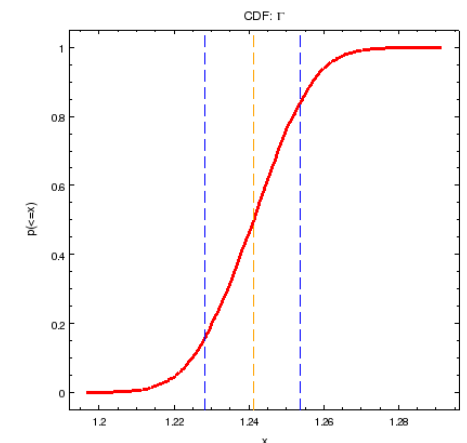
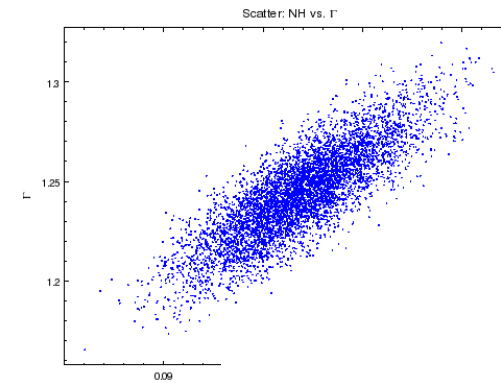
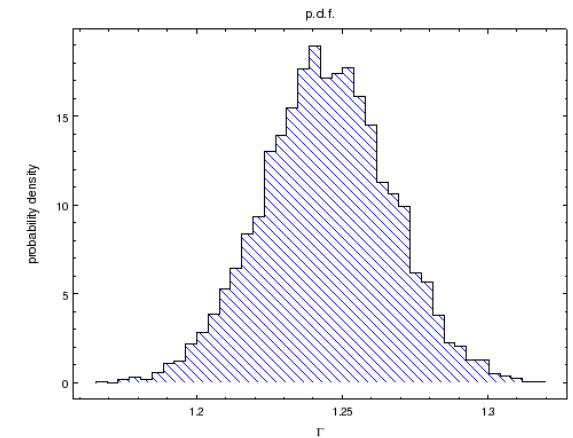


Sherpa: Modeling and Fitting in Python



Sherpa

- comes with well-tested, robust optimization methods - e.g. Levenberg-Marquardt, Nelder-Mead Simplex or Monte Carlo/Differential Evolution
- comes with statistics for modeling Poisson or Gaussian data
- can perform Bayesian analysis with Poisson Likelihood and priors, using Metropolis or Metropolis-Hastings algorithm in the MCMC (Markov-Chain Monte Carlo); allows to include non-linear systematic errors (calibration uncertainties) in the analysis
- is extensible (with python and compiled code):
 - is used in CIAO tools and scripts
 - in the Xija Chandra thermal modeling code
 - is used in the TeV HESS data analysis software
 - is used in the IRIS spectral energy distribution program

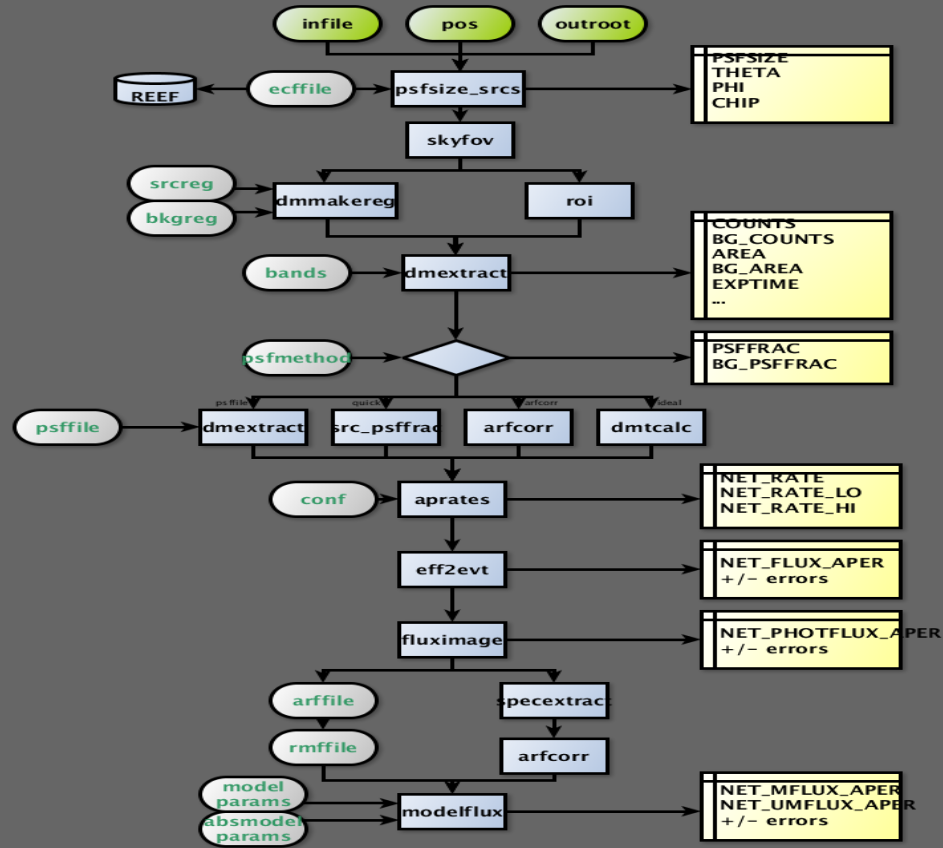


CALDB

- The CALDB (Calibration Database) contains everything you need that's not part of your specific observation.
- It's designed as a multimission directory structure. The Chandra files are in `$CALDB/data/chandra`
- Within that, they are arranged by instrument and kind of calibration. But, with luck, the software will find the CALDB files you need automatically.
- Just make sure that you keep the CALDB up to date! But, be careful - if you start off processing with a given version of the CALDB and CIAO, then upgrade to a new CALDB and CIAO, things are sometimes incompatible. Check the release notes.

Calculating Source Flux

```
$ srcflux myevt2.fits "03:29:29.250 +31:18:34.73" myflux
```



Encodes the logic described in six different CIAO threads.

Return count rates and fluxes and errors with all appropriate corrections.

Uses many tools written for the Chandra Source Catalog.

Complementary to it for special cases and fields not covered by the catalog.

Summary of source fluxes

Position	0.5 - 7.0 keV
	Value 90% Conf Interval
3 29 29.25 +31 18 34.7	Rate 0.0398 c/s (0.0381,0.0415)
	Flux 5.17E-13 erg/cm2/s (4.94E-13,5.39E-13)
	Mod.Flux 4.38E-13 erg/cm2/s (4.2E-13,4.57E-13)

srcflux capabilities

- finds auxiliary files automatically, like specextract
- automatically determines PSF-appropriate extraction region size for source and background, or accepts user choice
- uses one of four methods to apply aperture correction
- runs on multiple energy bands including named CSC bands
- accepts one position or a list (catalog of sources)
- calculates count rates using aprates method
- calculates fluxes two different ways (specified spectral model and eff2evt method; however, no spectral fit is performed)
- generates spectral responses for further analysis

Ongoing work: handling of warning flags for hard cases, e.g. chip edge