CIAO

Introduction and Scripts

Jonathan McDowell
Chandra X-ray Center, SAO
Scope

- Caveat: will cover ACIS imaging data only
- Basics the same for HRC and gratings, but with extra wrinkles
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)
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)
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)

The Chandra PSF

Off-axis angle

Pileup Effect

Specific instrumental effects eg.
- readout streak
- pileup - two or more photons detected as single event
Basics of CIAO

- Data files are in FITS format (usually binary tables, not just images)
- CIAO can also operate on ASCII file in many cases
- All (well, almost all) CIAO tool 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
dmclist infile opt=blocks,cols,data

ahelp dmclist → help for tool dmclist
plist dmclist → list parameters for dmclist

Key applications:
Sherpa - fitting
ChIPS – plotting
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

<table>
<thead>
<tr>
<th>CoNo</th>
<th>Name</th>
<th>Unit</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>time</td>
<td>s</td>
<td>Real18</td>
<td>154361559.612799964:154468627.4158599973 S/C TT corresponding to mid-exposure</td>
</tr>
<tr>
<td>2</td>
<td>ccd_id</td>
<td>Int2</td>
<td></td>
<td>0:9: CCD reporting event</td>
</tr>
<tr>
<td>3</td>
<td>node_id</td>
<td>Int2</td>
<td></td>
<td>0:3: CCD serial readout amplifier node</td>
</tr>
<tr>
<td>4</td>
<td>expno</td>
<td>Int2</td>
<td></td>
<td>0:2147483647: Exposure number of CCD frame containing event</td>
</tr>
<tr>
<td>5</td>
<td>chip(chipx,chipy)</td>
<td>pixel</td>
<td>Int2</td>
<td>1:1024: Chip coords</td>
</tr>
<tr>
<td>6</td>
<td>tdet(detx,tdeny)</td>
<td>pixel</td>
<td>Int2</td>
<td>1:8192: ACIS tiled detector coordinates</td>
</tr>
<tr>
<td>7</td>
<td>det(detx,deny)</td>
<td>pixel</td>
<td>Real4</td>
<td>0.50: 8192.50: ACIS detector coordinates</td>
</tr>
<tr>
<td>8</td>
<td>sky(x,y)</td>
<td>pixel</td>
<td>Real4</td>
<td>0.50: 8192.50: sky coordinates</td>
</tr>
<tr>
<td>10</td>
<td>pha_ro</td>
<td>adu</td>
<td>Int4</td>
<td>0:36855: total pulse height of event</td>
</tr>
<tr>
<td>11</td>
<td>energy</td>
<td>eV</td>
<td>Real4</td>
<td>0:1000000.0: total read-out pulse height of event</td>
</tr>
<tr>
<td>12</td>
<td>pi</td>
<td>chan</td>
<td>Int4</td>
<td>0:1024: nominal energy of event (eV)</td>
</tr>
<tr>
<td>13</td>
<td>fltgrade</td>
<td>Int2</td>
<td></td>
<td>0:255: pulse invariant energy of event</td>
</tr>
<tr>
<td>14</td>
<td>grade</td>
<td>Int2</td>
<td></td>
<td>0:7: event grade, flight system</td>
</tr>
<tr>
<td>15</td>
<td>status[4]</td>
<td>Bit(4)</td>
<td></td>
<td>0: event status bits</td>
</tr>
</tbody>
</table>
Energy slices through an event list, 0.1 - 10 keV
Level 1 Event List - Calibrated but Dirty

- Node boundaries
- Lots of background
- Bad columns
- Source!
- Bad pixels
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!
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 \textit{Exposure Map}, \( E(\Delta h, \lambda, \hat{p}) \), obtains 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 \([\text{phot cm}^{-2}\text{s}^{-1}\text{Å}^{-1}]\).

\textbf{Instrument Map} – efficiency calibration information, band integrated. (create with \texttt{mkinstmap})

\[
\text{mirror area} \times \text{detector QE}
\]

\textbf{Exposure Map} – applies telescope aspect history and coordinate transformations \((= \text{area} \times \text{time})\). (create with \texttt{mkexpmap}).

\[
\text{Instmap} \quad \text{Aspect}
\]
Problem 3: Exposure map is energy dependent; must assume a spectrum if using a broad band
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 $\Delta T$.

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 $x', y'$

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

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)

$$N(E'_i, x'_i, y'_i) = \int \int \int A(E)R(E, E'_i)F(E, x, y, t)P(x-x'_i, y-y'_i)q(E, x'_i, y'_i)dE dx dy dt$$
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 know 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

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

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

- 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
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.
MAKING X-RAY ANALYSIS EASIER

ciao_install - automated installation process
    - tunable, also supports source builds

What data is there? WebChaser is still great, but sometimes find_chandra_obsid is handy for CL use or scripting:

```
neptune>
neptune> find_chandra_obsid 'NGC 2403'
# obsid  sepn  inst  grat  time  obsdate  piname       target
2014   0.2    ACIS-S  NONE  36.0  2001-04-17  CAPPI  NGC2403
2937   2.6    HRC-I  NONE  2.8   2002-01-27  SUGIHO "NGC2403 S3"
4628   2.7    ACIS-S  NONE  47.1  2004-08-23  Lewin "SN 2004dj"
4629   2.7    ACIS-S  NONE  45.1  2004-10-03  Lewin "SN 2004dj"
4630   2.7    ACIS-S  NONE  50.6  2004-12-22  Lewin "SN 2004dj"
neptune>  
```

There's also the footprint service cxc.harvard.edu/cda/footprint

find_chandra_obsid can also download the data, or you can use...
download_chandra_obsid gets the data for you
This one makes subdirs 4628/ and 4629/ each with the usual primary/, secondary/ subdirs that you are used to
Next we update the archive processing with the latest calibrations using `chandra_repro`

```
neptune> ls 4628
axaff04628N002_VV001_vv2.pdf  oif.fits  primary/  secondary/
neptune> chandra_repro
Input directory (./): 4628
Output directory (default = $indir/repro) ():
```

Now we have a new `repro` subdirectory with (hopefully) all the files you'll need for further analysis, including “repro_evt2.fits”

```
neptune> ls 4628
axaff04628N002_VV001_vv2.pdf  oif.fits  primary/  repro/  secondary/
neptune> ls 4628/repro
acisf04628_000N003_bpix1.fits  acisf04628_000N003_stat1.fits  acisf209642202N003_pbk0.fits
acisf04628_000N003_fov1.fits  acisf04628_asol1.lis  pcadf209643885N003_asol1.fits
acisf04628_000N003_msk1.fits  acisf04628_repro_bpix1.fits  
acisf04628_000N003_mtl1.fits  acisf04628_repro_evt2.fits
```

`chandra_repro` also works on grating data
Now you have calibrated data and are ready to do science.
You may want to take a look at the data by making a three color fluxed image using 'fluximage'; cd into the repro directory and run as shown here.

- knows about CSC bands soft, med, hard, broad
- finds the asol, badpix, mask etc. on its own
- makes exposure maps etc.:
Adding four observations shows limitations of old script: obsid no 3 has a different SIM position and obsid 4 is a subarray; the new script handles the exposure maps and reprojection correctly in these cases. Avoid bad pixels at edge with thresholding.
merge_obs – Summary.

The new script

- parallelizes the computation across multiple processors on the host machine
- automatically determines the center and size of the mosaic (if the user doesn't specify)
  by averaging the unit vectors of the pointing directions and taking the union of the reprojected field-of-view polygons
- modifies headers to account for the fact that the 'sky' pixel coords go beyond their normal range (which can cause ds9 not to display part of the image)
- automatically handles different event input formats by trimming columns as needed
- automatic location and use of mask, aspect, bad pixel, parameter block files using values seeded in event file header
- sorts input files in time order
- for HRC-I, subtract particle background model
- thresholds final image using exposure map (default 1.5% of max exposure)
- cleans up intermediate files on exit
- supports standard catalog energy bands e.g. 'CSC', 'soft' as well as user-specified ones; can use spectral weight files for exposure maps if supplied

Limitations:
  Cannot combine ACIS with HRC-I/S, or HRC-I with HRC-S
  No ACIS background subtraction
  No support yet for improving astrometry before merging
Combining Observations – Example 2

Eta Carina

Raw counts (left)

Exposure map (right)
Combining Observations – Example 2 cont

Eta Carina

40 ACIS-I datasets 1999-2008

Mix of FAINT and VFAINT

Exposure times from 10 to 90 ks

Input was simple list of event files:

```
ls */*evt2* > lis
merge_obs @lis"[ccd_id=:3]" out
```

Result is a set of 1363 x 1537 pixel images (size autocalculated to cover the field)
Grating data

**chandra_repro:**
- extracts PHA2 file
- recent mod to retain manal V&V extraction region rather than overwrite
- plan to enhance to include responses for each arm and order

**tgextract2**
- extract spectra with customized source, bkg extraction regions
- useful for multiple source case

**combine_grating_spectra**
- coming soon, will coadd spectra and weight responses for
  - multiple orders
  - multiple exposures

**tg_findzo:**
- methods to find zero order pos even when center is blanked or piled

**TGCAT** (Huenemorder et al)
- tgcat.mit.edu
Processed grating archive -
  manually optimized extraction regions
  extractions for almost all grating observations
  high level extracted properties
Fluxes

**specextract** -
Source and background ACIS spectra for point and extended cases
- Weighted or unweighted ARF and RMF, grouped spectra
- BUT: still sometimes awkward to use

will improve to automatically locate auxiliary files if chandra_repro has been used

**combine_spectra**
- sum multiple imaging PHA spectra, responses
  (better to do independent fits but more convenient at low S/N)

*new* higher level script **srcflux** which wraps use of several existing CIAO tools and scripts

**srcflux** evt2.fits ra,dec src.out

- generate regions using typical psf size
- use **aprates** to determine count rates and confidence intervals (or upper limits)
- run specextract to generate responses
- use **eff2evt** to estimate fluxes
- use **modelflux** to estimate fluxes given spectral model
What is the X-ray flux of this source?
Calculating Source Flux

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

<table>
<thead>
<tr>
<th>Position</th>
<th>Value 90% Conf Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 29 29.25 +31 18 34.7</td>
<td>Rate 0.0398 c/s (0.0381,0.0415)</td>
</tr>
<tr>
<td></td>
<td>Flux 5.17E-13 erg/cm2/s (4.94E-13,5.39E-13)</td>
</tr>
<tr>
<td></td>
<td>Mod.Flux 4.38E-13 erg/cm2/s (4.2E-13,4.57E-13)</td>
</tr>
</tbody>
</table>
% srcflux repro/acisf06436_repro_evt2.fits "03:29:17.653 +31:22:44.97" mysrc
srcflux

infile = repro/acisf06436_repro_evt2.fits
pos = 03:29:17.653 +31:22:44.97
outroot = mysrc
bands = broad
srcsrc =
bkgreg =
bkgresp = yes
psfmethod = ideal
psffile =
  conf = 0.9
rmffile =
arffile =
  model = xsphabs.abs1*xspowerlaw.pow1
paramvals = abs1.nH=0.0;pow1.PhoIndex=2.0
absmodel =
absparams =
  abund = angr
fovfile =
asolfile =
pbfile =
mskfile =
bpixfile =
ecffile = CALDB
parallel = yes
nproc = INDEF
tmpdir = /tmp
clobber = no
verbose = 1
mode = ql

- echoes param choices
Extracting counts
Setting Ideal PSF : alpha=1 , beta=0
Getting net rate and confidence limits
Getting model independent fluxes
Getting model fluxes
Getting photon fluxes
Running tasks in parallel with 4 processors.
Running eff2evt for mysrc_broad_0001_src.dat
Running aprates for mysrc_broad0001_rates.par
Running eff2evt for mysrc_broad_0001_bkg.dat
Making response files for mysrc_0001
Running modeflux for region 1
Adding net rates to output
Appending flux results onto output
Appending photflux results onto output
Computing Net fluxes
Adding model fluxes to output
Scaling model flux confidence limits

---

Summary of source fluxes

<table>
<thead>
<tr>
<th>Position</th>
<th>Value</th>
<th>90% Conf Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 29 17.65 +31 22 44.9</td>
<td>0.0609 c/s (0.0587,0.063)</td>
<td></td>
</tr>
<tr>
<td>Rate</td>
<td>5.43E-13 erg/cm2/s (5.24E-13,5.62E-13)</td>
<td></td>
</tr>
<tr>
<td>Flux</td>
<td>5.88E-13 erg/cm2/s (5.67E-13,6.08E-13)</td>
<td></td>
</tr>
</tbody>
</table>
- 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