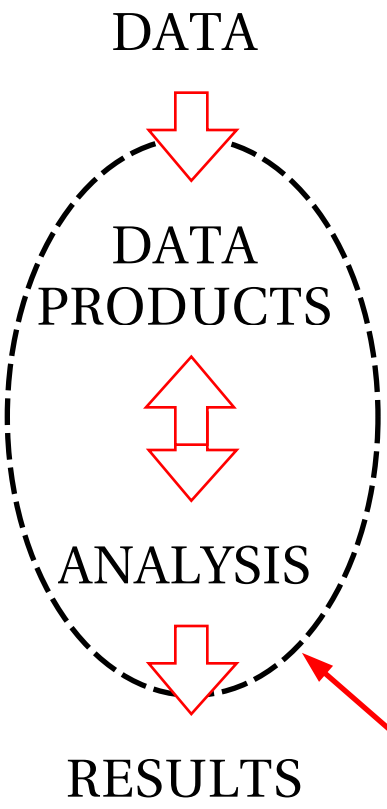
Introduction to CIAO



- Event List + Ancillary Files

- Recalibration
- Image
- Spectrum
- Lightcurve
- ...

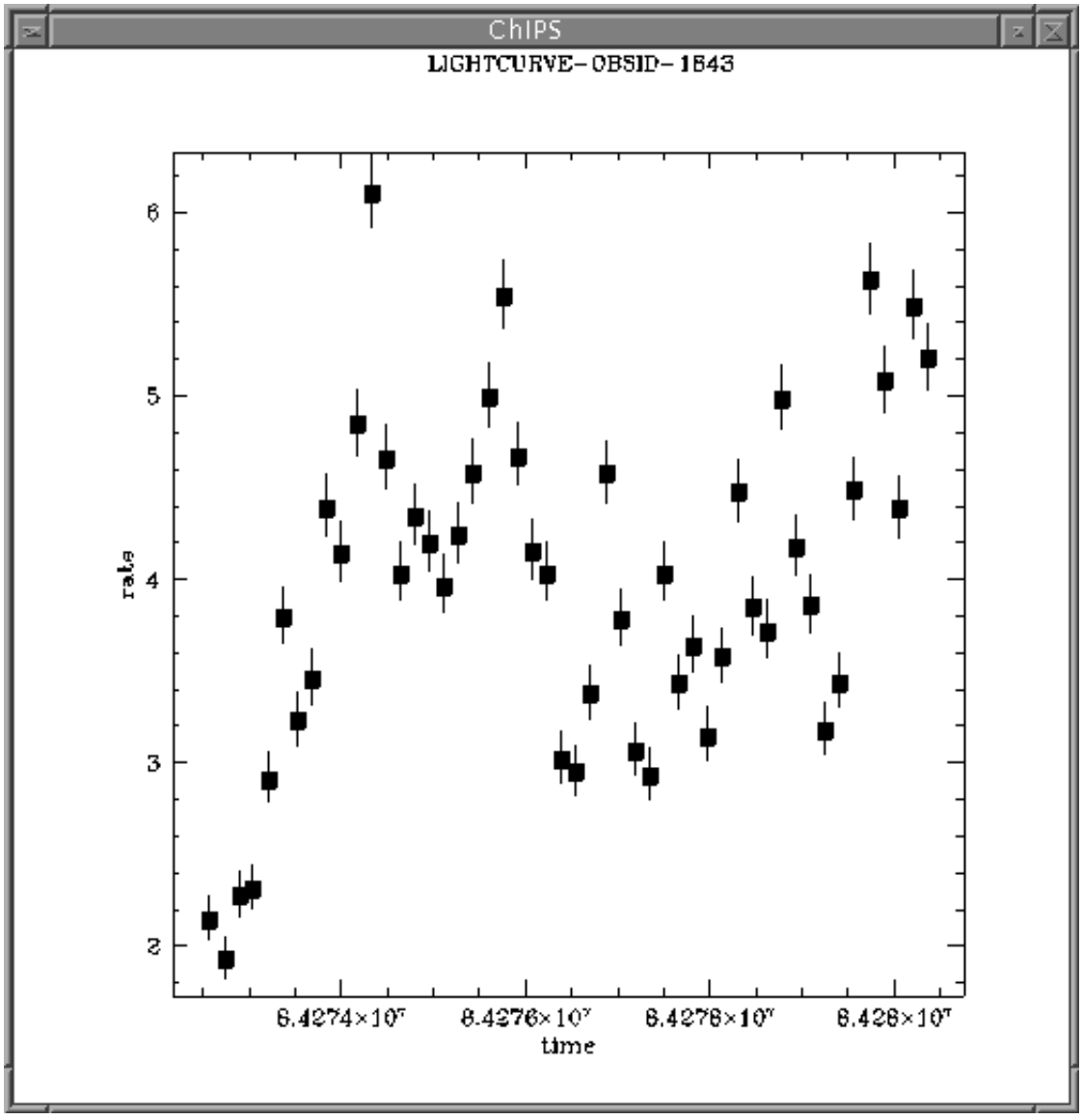
- Source Detection
- Spectral Fitting
- Power Spectrum Analysis

CIAO + CALDB + Threads and Scripts

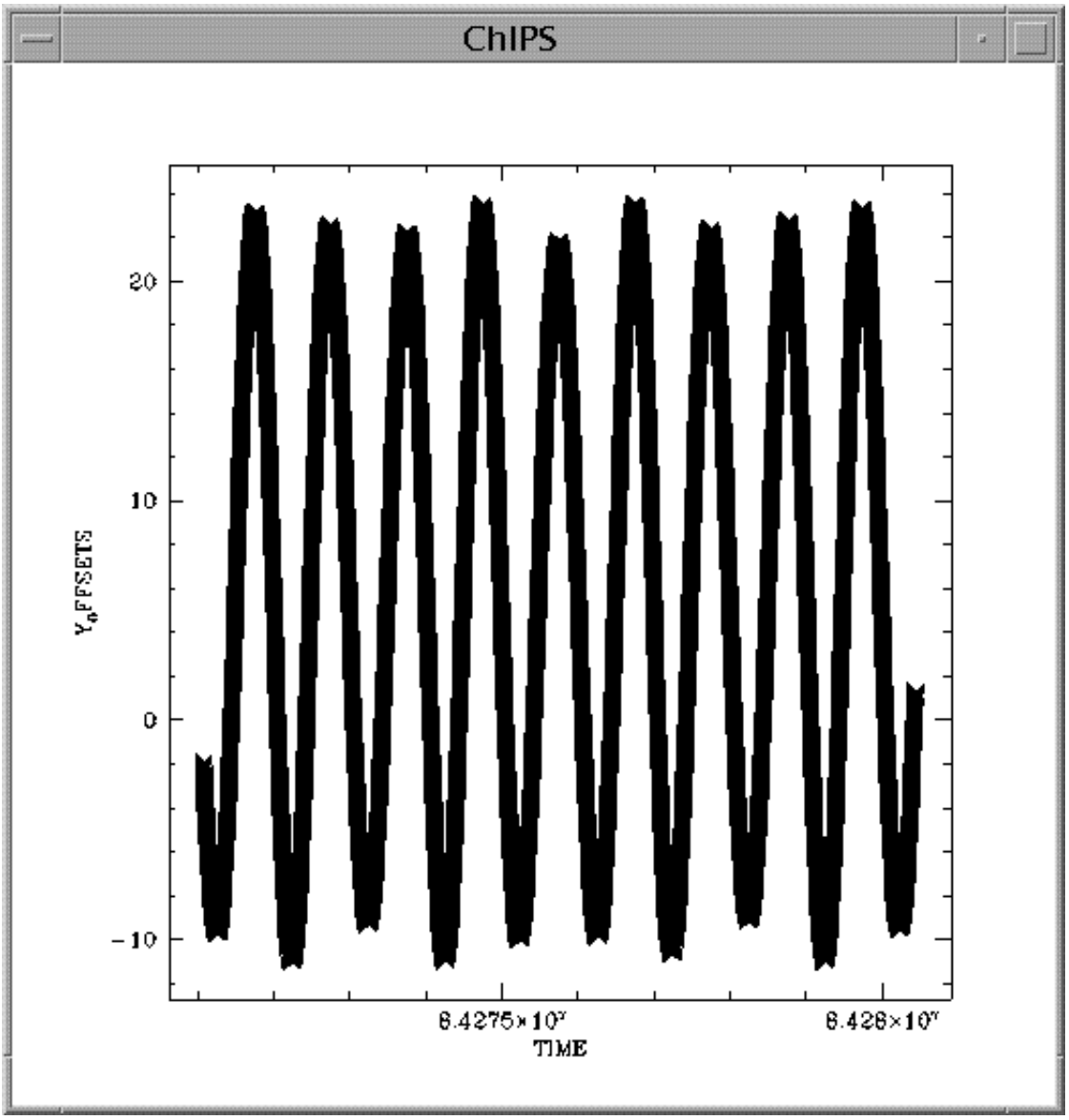
Although CIAO contains Chandra-specific programs, the majority of the tools can work on data from any instrument.

An example analysis session:

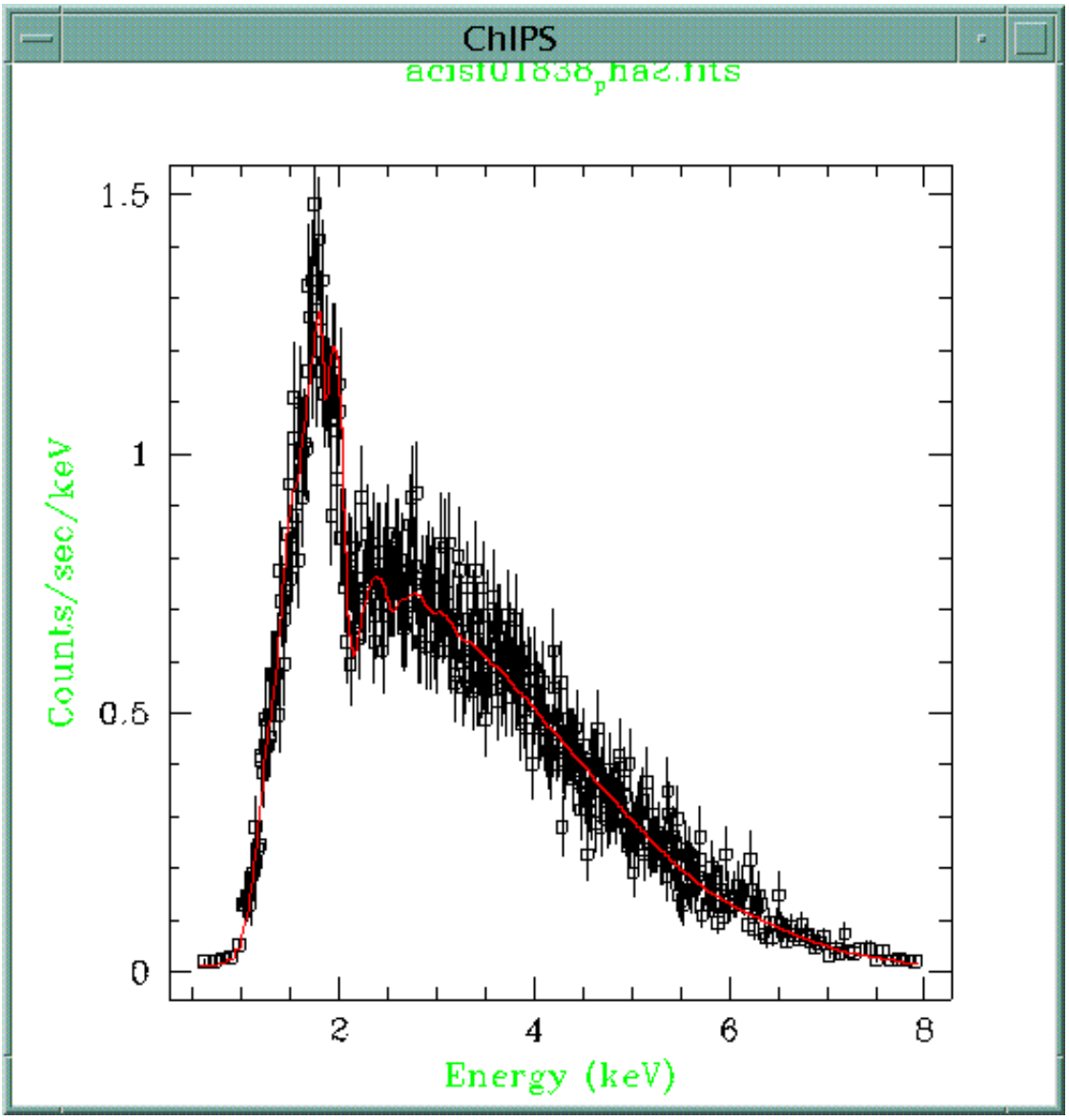
- filter the data
  - lightcurve, dmgti, dmcopy
- create aspect data
  - asp_apply_sim, asphist
- source detection
  - cell-, wav-, or vtpdetect
- extract spectra
  - dmextract
- calculate response
  - mkrmf, mkarf
- fit models
  - sherpa
- line identification
  - GUIDE



Examining the background lightcurve using firstlook



Plotting columns of the aspect file using prism



The best-fit model of a source (sherpa)

What is CIAO?

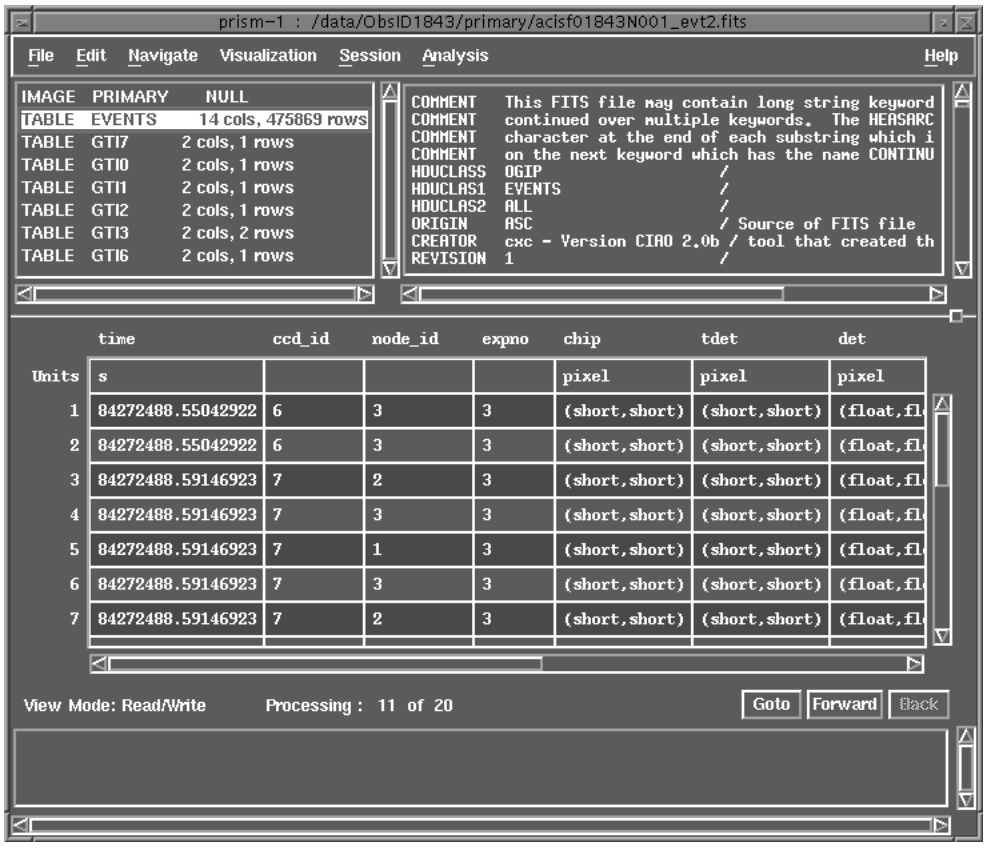
It is a collection of programs (both "atomic" and complex) that can be run from the shell or via a GUI (unlike IRAF and MIDAS). It comes with a collection of libraries (and source code is available) if you want to write your own code.

- Solaris 2.6, 2.7, 2.8
- Red Hat 6.2
- Slackware 7.0
- Alpha (coming soon)

What does it work with?

Chandra data is stored in FITS format. This is the "natural" system to use in an analysis session, although the QPOE format is supported and others are under consideration (the plotting tools read and write ASCII files).

FITS headers contain a large number of keywords relevant to the data (the processing version and history, observation details, ...). A single file can have data in multiple blocks (e.g. GTI or a weight map), and tables can contain vector columns.



The contents of a level-2 events file as displayed by prism

The CIAO Environment

We recommend the use of an alias called "ciao" to start up the system: it sets up a number of environment variables and path assignments (the appearance of the GUI tools can be changed using the ~/.CXCdefaults file).

- ciao -v what version you are using
- ciao -i information on CIAO setup

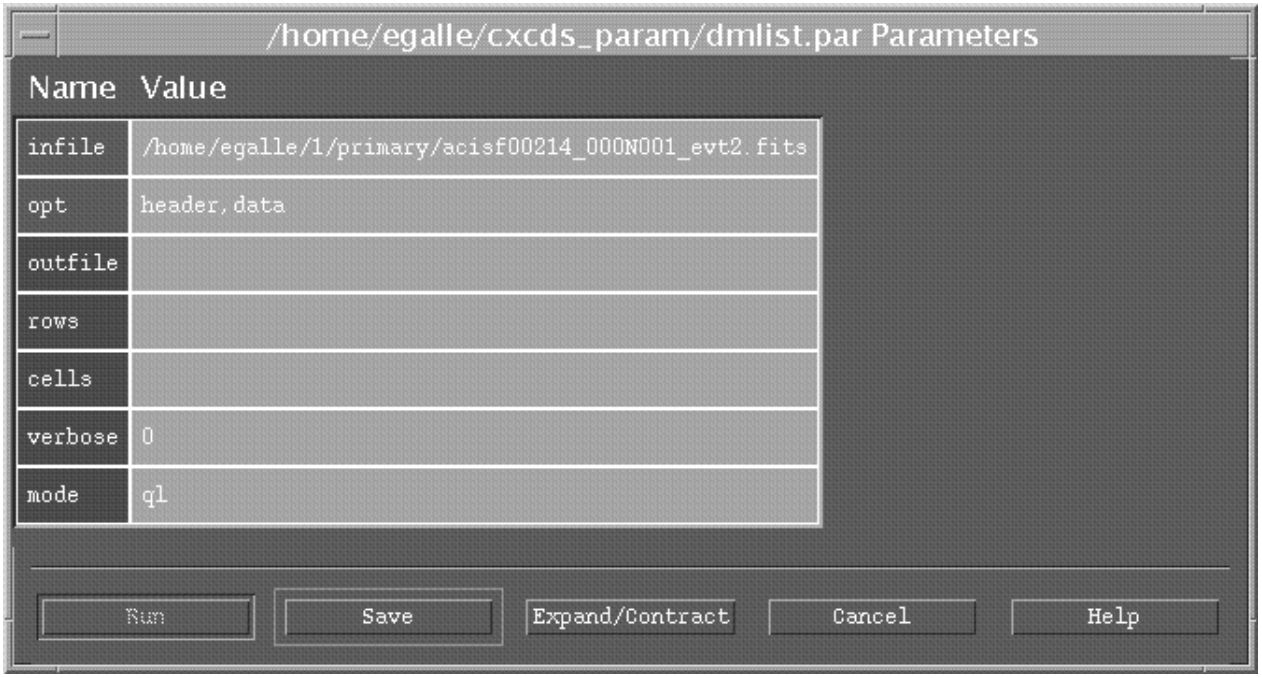
Parameter Files

Parameters to programs can be set on the command line or, as with IRAF and FTOOLS, using parameter files. These are stored in ~/cxcds_param/ by default, are called <tool>.par, and are ASCII files. A number of routines are provided to read and write to these files (e.g. plist, pset, punlearn).

These files provide a simple history mechanism, since you can see what you used the last time you ran the tool (e.g. GUIs). They are also useful for setting parameters that are not going to change during a session (e.g. a bad-pixel mask for an observation) - although care must be taken when handling more than one dataset!

```
unix% punlearn dmlist
unix% plist dmlist

Parameters for /home/janesmith/cxcds_param/dmlist.par

   infile =         Input dataset/block specification
      opt = data    Option
 (outfile = )       Output file (optional)
    (rows = )       Range of table rows to print
                    (min:max)
   (cells = )       Range of array indices to print
                    (min:max)
 (verbose = 0)      Debug Level(0-5)
    (mode = ql)


unix% pset dmlist infile=acisf01843N001_evt2.fits
unix% pset dmlist rows=1:2
unix% plist dmlist

Parameters for /home/janesmith/cxcds_param/dmlist.par

   infile = acisf01843N001_evt2.fits Input
                    dataset/block specification
      opt = data    Option
 (outfile = )       Output file (optional)
    (rows = 1:2)    Range of table rows to print
                    (min:max)
   (cells = )       Range of array indices to print
                    (min:max)
 (verbose = 0)       Debug Level(0-5)
    (mode = ql)
```

See ahelp parameter for more information (tab completion, redirection, mode values)



Editing a parameter file using the toolagent GUI (available via the *Analysis* menu of prism & filtwin)

Getting help

CIAO comes with its own help system called ahelp. All the tools have their own help text, as well as a number of other subjects such as parameter files, the ardlib, coordinate system (coords), and the data model (dm). Each file contains a list of associated help files - to aid browsing - and is also available on the Chandra web site (Documents/Ahelp).

Additional capabilities are provided on the command line:

- ahelp [-s|-m|-l] [subject][..context]
- ahelp -k keyword

The web site contains more resources: the dictionary (Documents/Dictionary), Frequently Asked Questions (Documents/CIAO FAQ), Manuals (Documents/Manuals and Advanced/Documents), and the Help Desk for when all-else fails. Also look at the *Threads* page: Documents/Threads.

There are also the Chandra Software Exchange - where users can contribute their own software - and the Chandra Users' Discussion Group (an email list).

Filters, Regions, and GTIs

This is an essential part of X-ray data analysis: the removal of information from an event list. It may be a "benign" operation (such as excluding a column or binning to produce an image) or the removal of a set of events (e.g. to remove periods of high background or poor aspect solution). Filters can be supplied as part of a filename, in which case the operation is performed "on the fly":

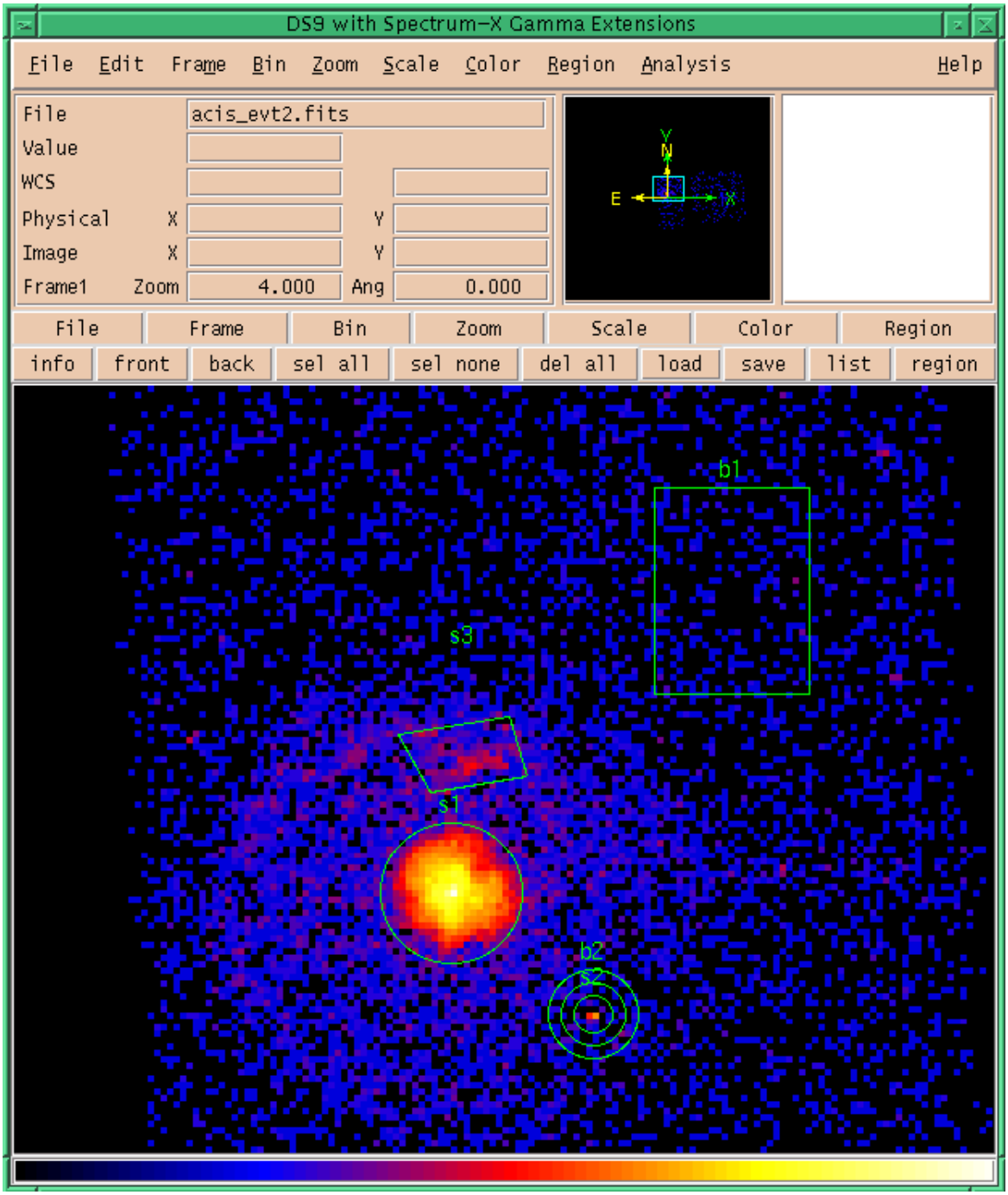
```
dmstat "evt2.fits[EVENTS][energy>300][cols -grade]"
```

See: ahelp dmfiltering

- GTIs (Good Time Intervals) are used to define what times periods of the observation can be used (i.e. contain valid data). They are generally stored as a block in the event list.

- Regions are used to define the source and background areas of an image. They are text files that can be created manually or within ds9, and are used as a filter (e.g. "[sky=region(source.reg)]").

The filters applied to a file are stored in its "subspace"; dmlist can read this history with opt=subspace.



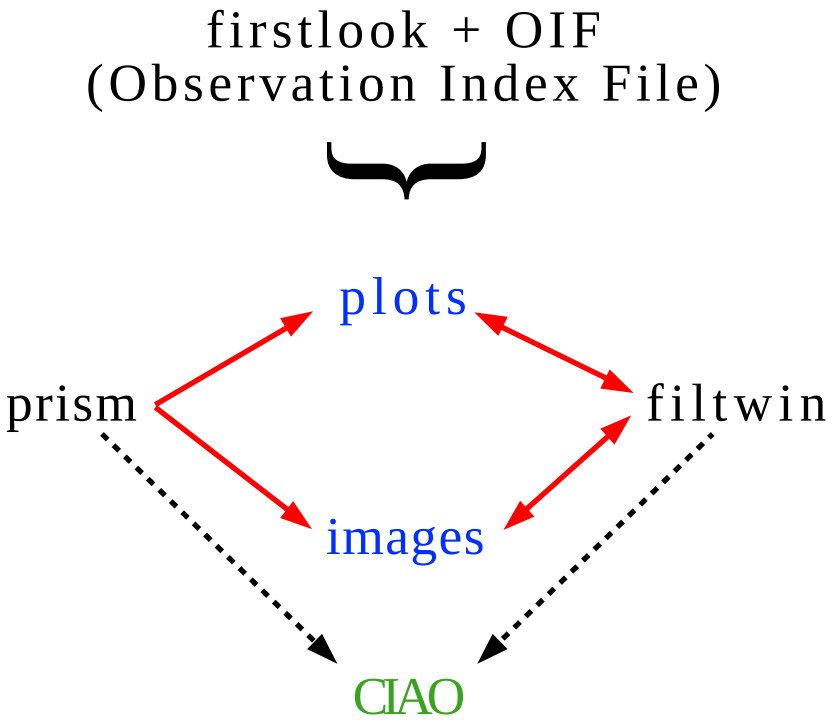
Several of the available regions as displayed by DS9

Overview

Datamodel
 copy, filter, extraction, stats, ...
Chandra Specific
   Instrument tools: update calibration, correct for instrumental effects, find & extract grating data, create aspect histograms
   Response tools: exposure map, PSF, RMF and ARF
Source Detection
   celldetect, wavdetect, vtpdetect
Timing & Background tools
 lightcurve, axbary, get_src_region
Convolutions, Transforms, & Smoothing
 csmooth, aconvolve, acrosscorr, apowerspectrum
Plotting (*)
   ChIPS
Modelling/Fitting (*)
   Sherpa (like XSPEC, but not restricted to spectral data)
Spectral Line Identification
   GUIDE

(*) powerful data manipulation and scripting capabilities are now possible with the inclusion of the S-Lang interpreted language.

CIAO in use



- prism

  An easy way to examine data files and produce images and plots

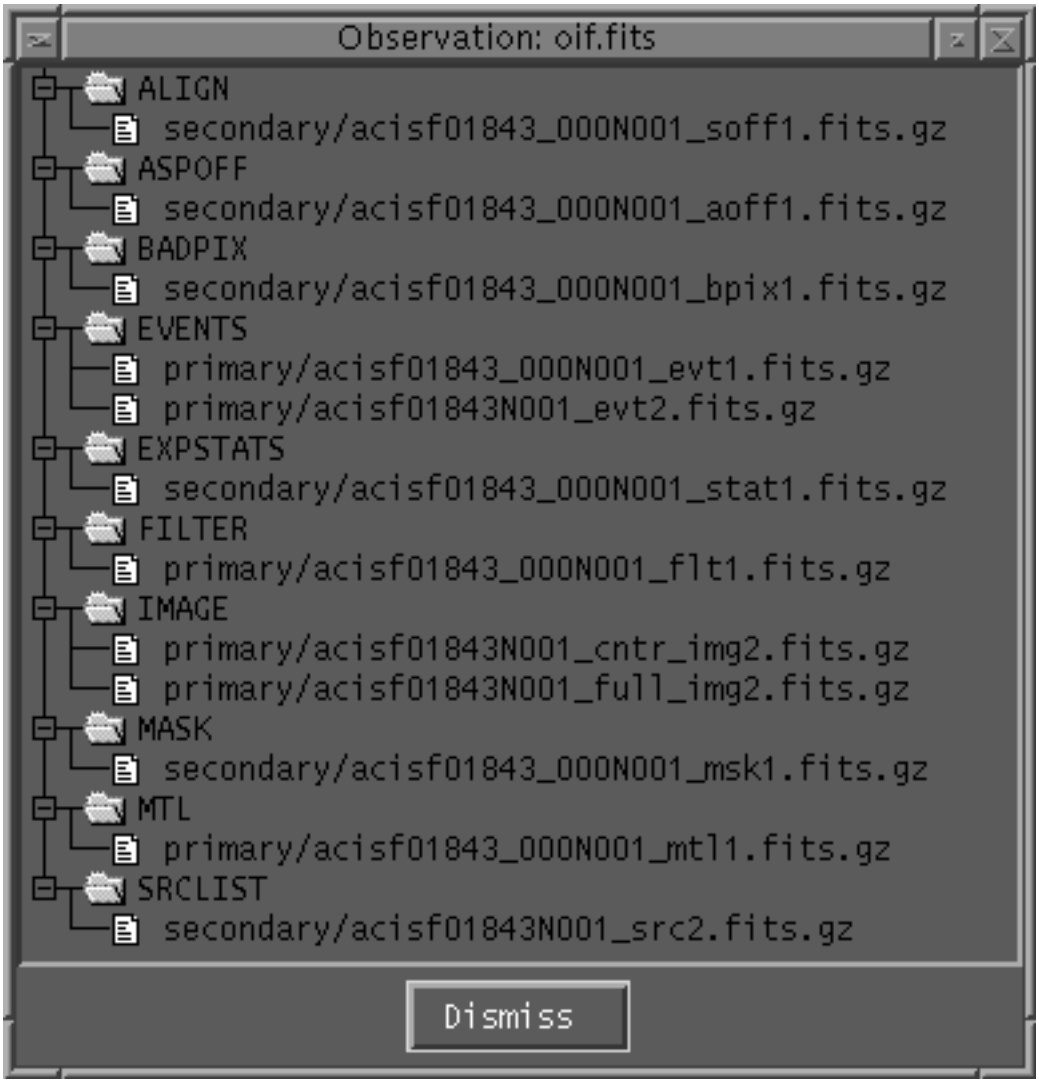
- filtwin

  Interactive filtering of a dataset

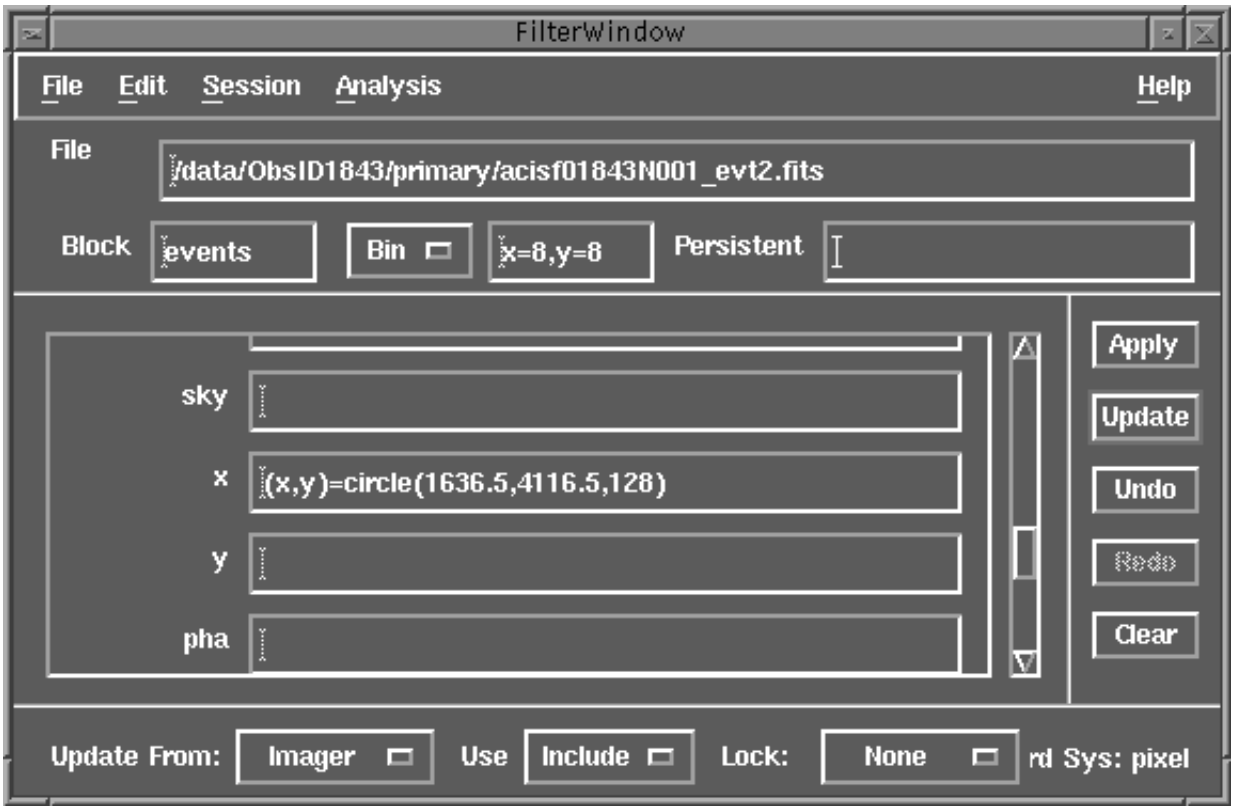
- firstlook

  Create: image, spectrum, lightcurve

The GUI tools are useful for exploratory data analysis. For the most flexibility use the command-line versions.



The contents of an OIF as displayed by firstlook



Filtering an event file using the filtwin GUI (aka *filterwindow*)

The first few columns of a pha2 file displayed in prism

Putting it all together

Two important web resources are the Threads (Documents/Threads) and Scripts (Download/Scripts) pages. Threads are our "how-to" documents, and provide a *step-by-step guide to common tasks*, while the scripts are written to automate certain tasks. These pages - like the whole site - are often updated, so it is worth visiting them regularly.

Current thread sections:

Introduction
: CIAO and basic tool use

Data Preparation
: Clean data, correct for problems

Imaging
: Combine data, source detection, exposure maps, source profiles

Imaging Spectroscopy
: Extract spectra and response data

Grating Spectroscopy
: Handle PHA2 data

Sherpa
: Fitting, and fake-ing, data

Reprocessing

The threads are designed to work with data processed using recent calibration data (so-called "reprocessed" data), so be careful if using older data. The estimated completion for reprocessing of old data is April 2001. The web pages contain information on the differences between the various processing versions.

CALDB

Much of the interaction with the calibration database is now hidden and automatic (e.g. the setting of gain or QE maps), although it can be over-ridden if required. The main times a user will interact with the CALDB is when making a RMF (which needs a Fits Embedded Function file), or when querying the PSF library.

S-Lang

S-Lang has been added to ChIPS and Sherpa, and provides a programming language with which to manipulate your data.

```
chips> # read data5.dat into 'fit' structure:
chips> fit = readfile("data5.dat")
chips> # show contents of fit:
chips> print(fit)
filename          =  data5.dat
path              =  /data/chips/
filter            =  NULL
ncols             =  2
nrows             =  12
col1              =  Float_Type[12]
col2              =  Float_Type[12]
chips> # find rows which are > 0:
chips> index = where( fit.col1 > 0 )
chips> # create xf to contain these rows:
chips> xf = fit.col1[index]
chips> # we can also use index to get the y values:
chips> yf = fit.col2[index]
chips> # print the data to the screen:
chips> writeascii(stdout,index,xf,yf)
1       0.5     1.81205
2       1.5     2.25558
3       2.5     2.6746
4       3.5     3.05986
5       4.5     3.40211
6       5.5     3.6921
7       6.5     3.92057
8       7.5     4.07828
9       8.5     4.15598
10      9.5     4.14442
11      10.5    4.03435
```