ANNOUNCEMENTS

The website for the workshop is
http://cxc.harvard.edu/ciao/workshop/sep19/

Quick way to get there…
cxc.harvard.edu —> Data analysis —> Workshops

You can find there

- All the presentations
- A workbook with Exercises for the hands-on session

- If you need a certificate of attendance let us know and we can email you one

An Introduction to CIAO

(Chandra Interactive Analysis of Observations)

ChIPS*

* Development frozen as of CIAO 4.11 - Moving toward Matplotlib - Conversion guide provided

CIAO: Chandra's data analysis system
Fruscione et al. 2006, SPIE Proc. 6270, 62701V, D.R. Silvia & R.E. Doxsey, eds.

What is CIAO?

Why are we all here (on a Saturday and Sunday!)?

First a linguistic note….

CIAO …..from "s'sciavo", "I am your servant" in Venetian language

And CIAO has been at the service of X-ray astronomers for 20 years!

From: http://cxc.harvard.edu/ciao/ahelp/ciao.html

AHELP for CIAO 4.11 **ciao** **Context: concept**

Synopsis

Chandra Interactive Analysis of Observations

Description

The remarkable science capabilities of the **Chandra X-ray Observatory** demanded **new**, **flexible**, **multi-dimensional**, software to analyze the data it returned. The result is CIAO - the Chandra Interactive Analysis of Observations - a system that has proven itself useful for the **analysis of data from other, non-X-ray missions**, because of the **mission independence** that is the basis of the CIAO design.

Introduction to CIAO Tools

CIAO is a data analysis system

written for the needs of users of the Chandra X-ray Observatory. Because Chandra is the first mission with 4-dimensional data (2 spatial, time, energy) in which each dimension has many independent elements, CIAO was **built to handle N-dimensional data** without concern about which particular axes were being analyzed. Also, **apart from a few Chandra instrument tools, CIAO is mission independent.**

In order to allow users of Chandra data to use pre-existing tools, all **CIAO tools read and write several formats**, including FITS tables (which includes event files), ASCII formats, and FITS images.

CIAO also needed to be able to **filter down and project the 4-D Chandra** event data to manageable sizes and convenient arrays. This has to be done flexibly, so we have built all CIAO tools to take a `**filtering and binning**' specification on the command line, making use of a general purpose `**regions**' syntax. See "ahelp dm" for information on the Data Model that makes all this possible.

Since Chandra data can be sliced and diced in so many ways, and because the Chandra calibration is spatially and energy dependent, we needed to **keep track of just how the data had been filtered and binned,** which we do in a `data subspace'. The tools keep track of this subspace automatically and allow users to review previous data processing. See "ahelp subspace".

The CIAO design allows **close interconnection of tools**. For example, the output of any of the source DETECT programs can be fed into dmextract to create a summed spectrum which can then be fit in Sherpa.

Modeling, Fitting, and Plotting

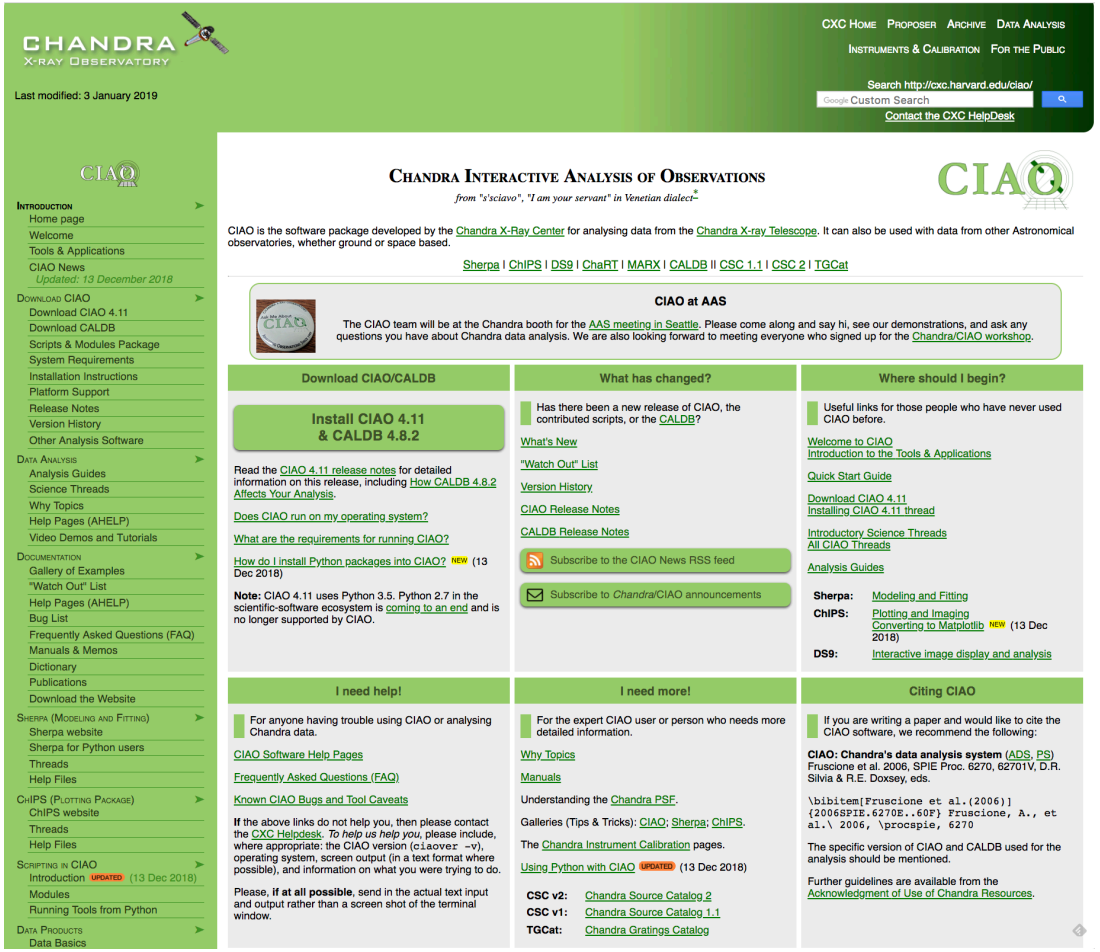
The **modeling and fitting tool Sherpa is central to the CIAO system**. Sherpa performs forward fitting of models to data in N-dimensions. Refer to "ahelp sherpa" and the Sherpa website for complete details.

ChIPS, the CIAO plotting package, can be used during data analysis - e.g. to plot a lightcurve - and to create publication-quality figures. ChIPS is designed for use in a variety of modes: as a user-interactive application and in batch mode. The Sherpa application uses ChIPS to display data fits, residuals, and so on. Refer to "ahelp chips" and the ChIPS website for complete details.

Altogether CIAO is a **powerful system for the analysis of all types of data.**

What is CIAO? …. In practice

- A collection of Unix command line tools and Python applications
- Shares code with standard Chandra processing pipeline
- Allows Chandra instrument specific data reduction (eg ARF and RMF)
- Tailored to specialized X-ray astronomy data analysis, but not specific to Chandra (as been used with XMM)
- Coded with attention to standards and interoperability so that generic tools can be (and are) used for other X-ray data and even optical and radio data (e.g. multiwavelength analysis)
- Easy for beginners, yet powerful for advanced users
- Linux and Mac, annual releases
- Installed 1500+ times per year (single users to large institutions)



General Concepts

1. File format
2. Parameter Files
3. Filtering and Binning (the Data Model)
4. Regions
5. Subspace
6. Good Times Intervals (GTI)
7. Scripting language (Python)

1. File Format

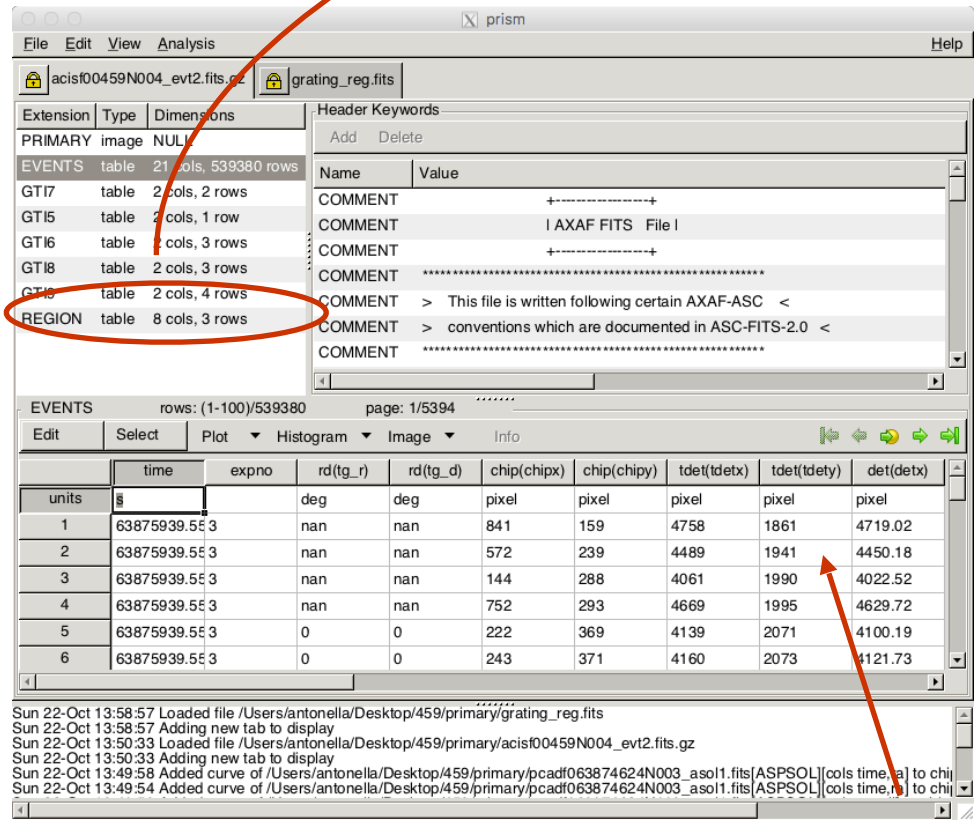
- Chandra data is stored in the (binary) **FITS format** however simple **ASCII (text) files** can be handled by many tools and applications.
- When CIAO operates on data it stores **processing state** and **processing information** along with the data (**keywords**, **subspace**).
- A single Chandra file can contain **multiple "datasets"** (e.g. data, Good Time Intervals, weight map, regions) which are **stored in "blocks"**.
- Blocks can contain image or table data.
- **dmlist** (a command line tool) or **prism** (a GUI) are available to view file contents.

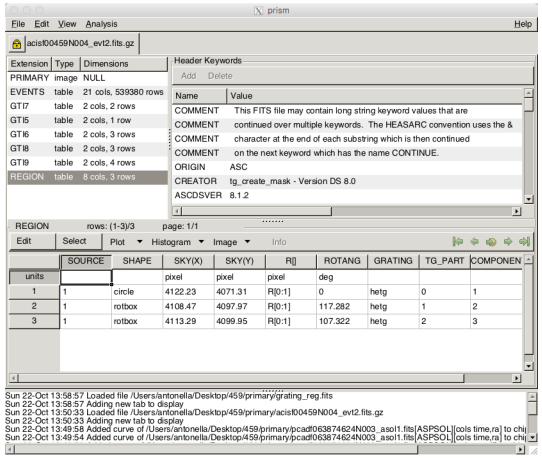
PRISM: file viewer, editor, tool launcher, quick plots

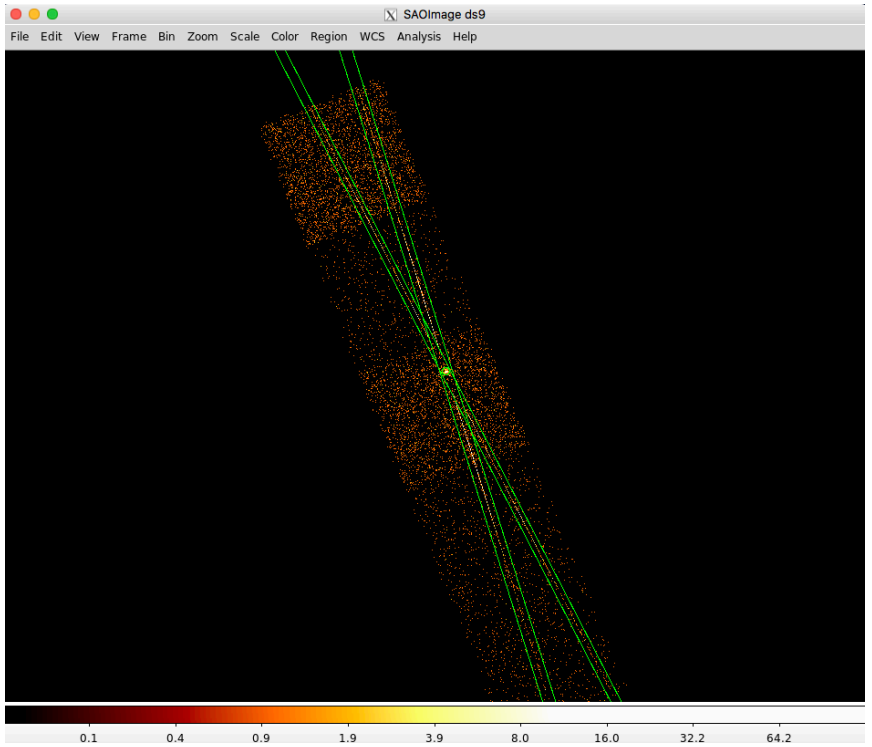
blocks

header

data







2. Parameter Files

From: http://cxc.harvard.edu/ciao/ahelp/parameter.html

- The CIAO tools use ASCII parameter files to get and store processing parameters (eg **dmcopy.par, specextract.par**)
- The interface is similar to the IRAF and FTOOLS systems

CAVEAT: We recommend that you start FTOOLS before CIAO to minimize conflicts (see http://cxc.harvard.edu/ciao/threads/ciao_startup/index.html#ftools)

- Parameters can be set via:
  - a number of routines (eg **punlearn, pset, plist**)
  - a "Parameter Editor GUI" (peg) (see **ahelp peg**)
  - the Python interface to the CXC parameter system (see **ahelp paramio**)
- Multiple values can be specified for a parameter if is listed in the help file as accepting stacks (see **help stack**)

Parameter file example

```
antonella% plist dmcopy

Parameters for /Users/antonella/cxcds_param4/dmcopy.par

        infile = acisf00459N004_evt2.fits.gz[REGION] Input dataset/block specification
       outfile = grating_reg.fits Output dataset name
       (kernel = default)         Output file format type
       (option = )                Option - force output type
      (verbose = 0)               Debug Level
      (clobber = no)              Clobber existing file
         (mode = ql)

antonella% dmcopy
Input dataset/block specification (acisf00459N004_evt2.fits.gz[REGION]):
Output dataset name (grating_reg.fits):
Clobber set to no, and output file grating_reg.fits exists.

antonella% pset dmcopy outfile=grating_reg2.fits
antonella% dmcopy

Input dataset/block specification (acisf00459N004_evt2.fits.gz[REGION]):
Output dataset name (grating_reg2.fits):

antonella%
```

Note required (infile, opt) and optional parameters (the ones in parenthesis)

3. Filtering and Binning (the "Data Model")

http://cxc.harvard.edu/ciao/ahelp/dm.html
http://cxc.harvard.edu/ciao/ahelp/dmfiltering.html
http://cxc.harvard.edu/ciao/ahelp/dmbinning.html
http://cxc.harvard.edu/ciao/threads/filter/

- Filtering (or the removal of unwanted events) is an essential part of X-ray analysis - e.g. to remove periods of high background or poor aspect solution, exclude uninteresting sources from an image etc.
- Binning is the action on event files which creates histograms, images, or 3D data cubes. It is useful for visualizing and manipulating data
- The "CIAO Data Model" (DM) is the versatile interface used by CIAO to examine and manipulate standard format datafiles (e.g. FITS, ASCII).
- The DM enables powerful filtering and binning
- The name "Data Model" reflects the fact that the interface can be used on data files of different format (all described by a single abstract description - the same "model") in a transparent way.

More About the Data Model

- An important characteristic of the DM is that **any** program that asks for a data file name as input accepts a ***"virtual file"*** string which causes the program to see a filtered version of the file in question **without the need to physically create a file on disk**

- The "virtual file" syntax is also commonly used to create a filtered version of the input file on disk

- Another important characteristic of the DM is that **all columns of event lists are treated "equally"**: for example binning is allowed not only in spatial coordinates but also in e.g. time, or energy coordinate, giving the ability of creating multidimensional images in space-energy, or space-time, etc.

Data Model Syntax (ahelp dmsyntax)

- All CIAO tools use the DM library and therefore accept as input “virtual files” described using the DM syntax.

- In the DM context a “virtual file” in represented by a filename followed by a series of optional qualifiers in square brackets [ ]:

“filename[block][filter][columns/binning][options][rename]”

where:

block - is the “section” of the file to use
filter- is the filter to be applied

columns/binning - specifies **either the columns** from a table to be included in an output table **or the binning**. When binning the data to generate an n-dimensional image, the range and binsize (min:max:bin) must be specified.

options - a sequence describing special options for the DM library

rename - specifies a name for the new block

- the order of the qualifiers generally matters, however...
- not all qualifiers need to be present always

Simple examples of "virtual files"

- A file which contains the first three columns of the EVENTS block specified by number:

(a) **acisf01843N001_evt2.fits[EVENTS][time=84245787:84247000][cols #1,#2,#3]**

- A file which contains the first three columns of the EVENTS block specified by name:

(b) **acisf01843N001_evt2.fits[EVENTS][grade=0,2,3][cols time,ccd_id,node_id]**

after the events are filtered in time or in grade

In the example above:

block: [EVENTS]
filter: [time=84245787:84247000] (a)
[grade=0,2,3] (b)
columns/binning:
[cols #1,#2,#3] (a)
[cols time,ccd_id,node_id] (b)

- An event file binned on the column called “PI” (to create a “PI spectrum”) for a specified region

acisf01843N001_evt2.fits[EVENTS][sky=region(mysrc.reg)][bin pi=1:1024:1]

(typical input to the tool dmextract)

- or binned in x,y coordinates to create an image

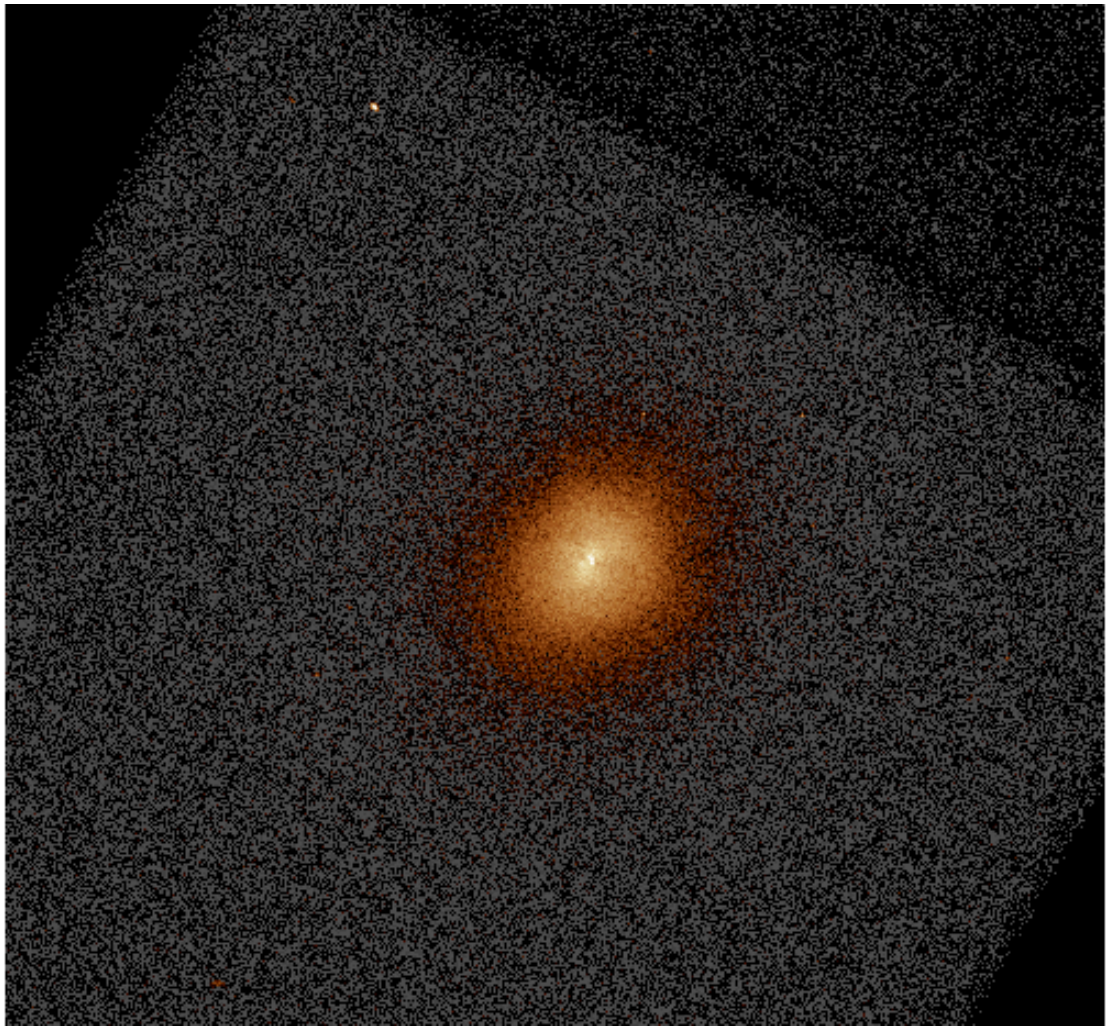
acisf01843N001_evt2.fits[EVENTS][energy<7000][bin x=320:480:4,y=320:480:4]

(typical input to the tool dmcopy)

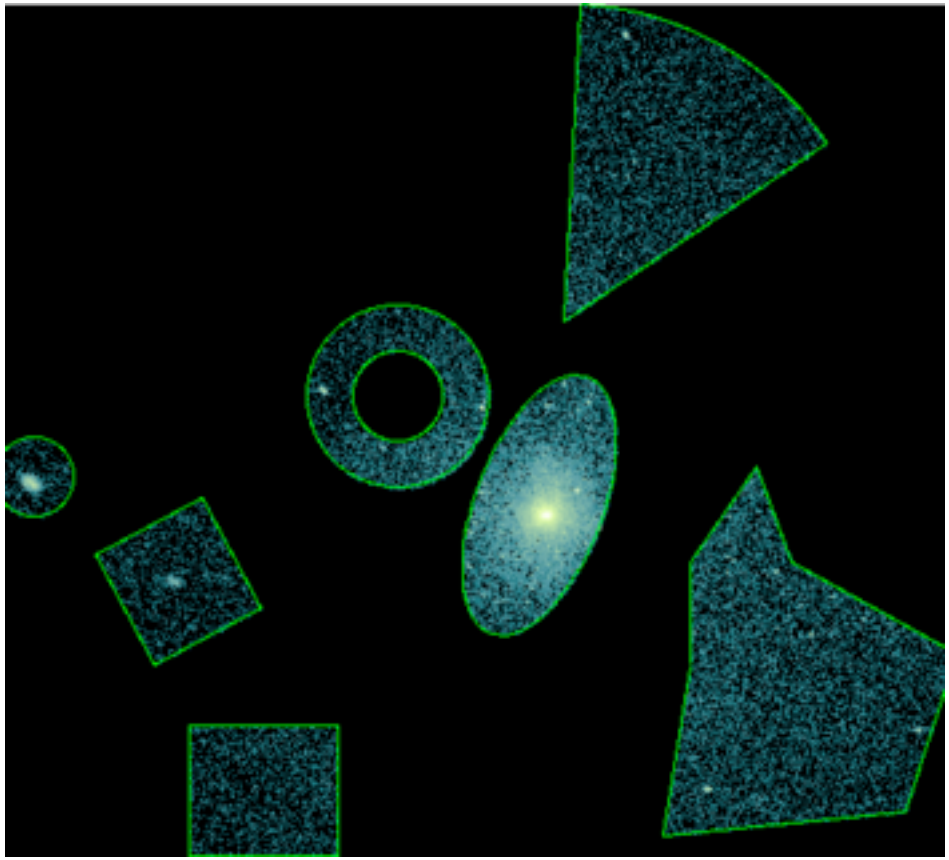
In the example above:

block: [EVENTS]
filter: [sky=region(mysource.reg)]
[energy<7000]
columns/binning:
[bin pi=1:1024:1]
[bin x=320:480:4,y=320:480:4]

```
dmcopy "acisf06934N002_evt2.fits[bin x=3500:4500:2,y=3500:4500:2]" 6934_sky_binsize.fits
```

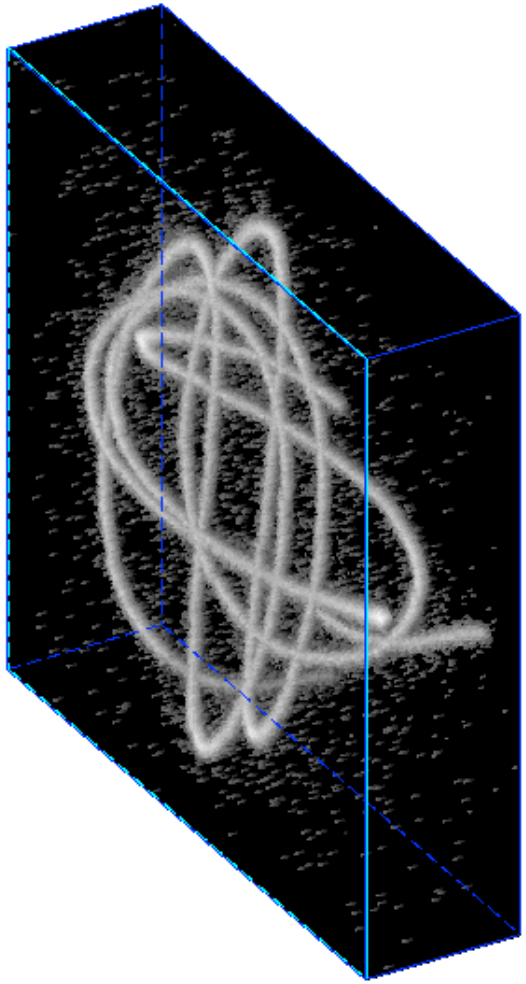


```
dmcopy "ngc1404.img[sky=region(ngc1404_sample.reg)]" ngc1404_regfile.img clob+
```



```
dmcopy "06540_evt.fits[(chipx,chipy)=box(8003.5,8137.5,512,512,0)]
[bin chipx=::2,chipy=::2,time=::#50]" outfile=cube.fits
```

This example shows a 3D image (cube) of a point source shown in chip coordinates. Since Chandra dithers during the observation, the point source moves across the detector versus time, which is the 3rd dimension.



Here an HRC event file is filtered on the `chip` coordinates and then binned into a cube. The X and Y axes are the chipx and chipy values binned by 2, and the third axis is time binned into 50 bins.

Data Manipulation Tools

The four DM “core” tools

dmlist: list contents or structure of a file

dmcopy: filter and bin tables and images

dmextract: make a histogram table file (e.g. PHA file, lightcurve file) from a table column. Generate count histogram on supplied regions for a spatial table or image file.

dmstat: compute statistics for images and columns in tables

30+ data manipulation tools are included in CIAO

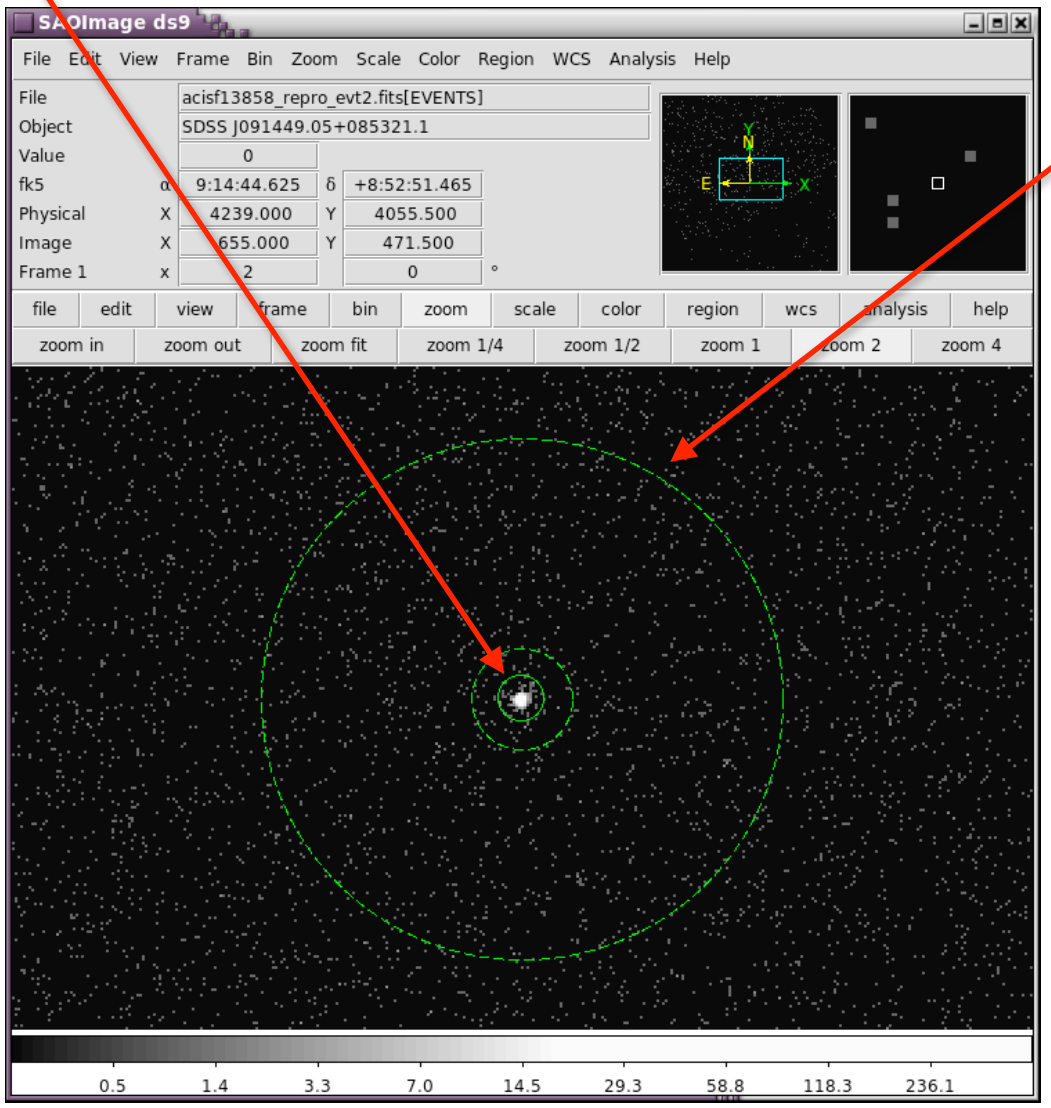
4. Regions

http://cxc.harvard.edu/ciao/ahelp/dmregions.html
http://cxc.harvard.edu/ciao/threads/regions/

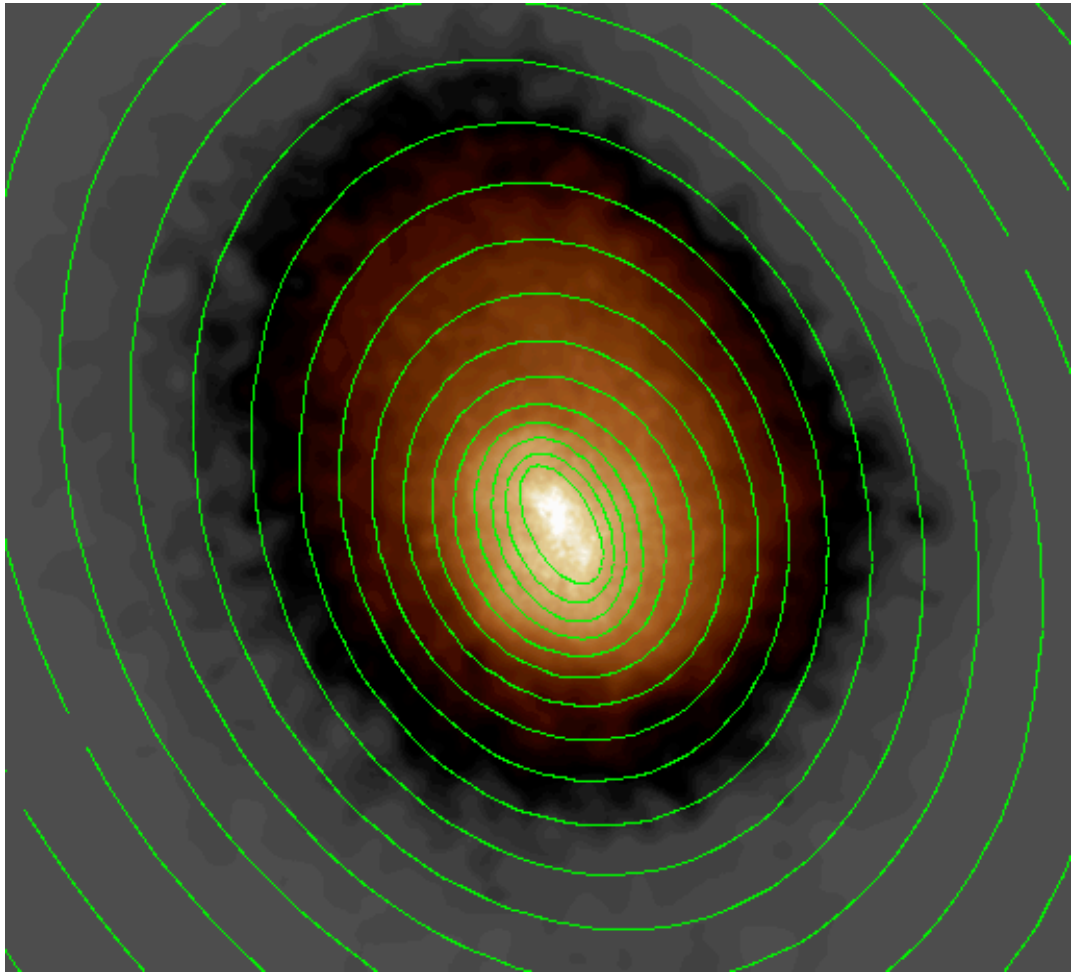
- Regions are **two dimensional filters** commonly used in CIAO data analysis to to include and/or exclude data
- For example regions are used to define the source and background areas in an image.
- They are text files or FITS files that can be created manually or within ds9, and can be used as a filter (e.g. “[sky=region(source.reg)]”)
- Regions can be combined using boolean AND (intersection) or boolean OR (union) operations.

```
circle(9:14:49.090,+8:53:21.231,4.083")
```

```
annulus(9:14:49.074,+8:53:20.987,9.064",46.425") # background
```



```
dmellipse a1664.asm a1664.ellipses "lgrid(0.1:0.96:0.05)" step=100 clob+
```



5. Subspace

From: http://cxc.harvard.edu/ciao/ahelp/subspace.html

- Subspace **records the filters applied to a file**
- Having this recorded in a data file header allows subsequent tools to extract appropriate calibration information. dmlist can read this history using opt=subspace

dmlist "acisf13736_evt2.fits[ccd_id=3,sky=circle(4324,3676,50)"] subspace

dmlist "acisf13736_evt2.fits[ccd_id=3,sky=circle(4324,3676,50)"] subspace

```
Data subspace for block EVENTS: Components: 1 Descriptors: 16
-------------------------------------------------------------------------------

--- Component 1 ---
  1 time                Real8                TABLE GTI3

                                             444947637.1627430916:444970563.5344673395
  […]
  3 ccd_id              Int2                 3:3
  4 node_id             Int2                 0:3
  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                 Real4                Circle(4324,3676,50)
  8 sky                 Real4                Field area = 6.71089e+07 Region area = 7853.98

  8 sky                 [ 1] x                    4274.0:     4374.0
  8 sky                 [ 2] y                    3626.0:     3726.0
  9 phas                Int2                 -4096:4095
 10 pha                 Int4                 0:36855
 11 pha_ro              Int4                 0:36855
 12 energy              Real4                                  0:  1000000.0
 13 pi                  Int4                 1:1024
 14 fltgrade            Int2                 0:255
 15 grade               Int2                 0:0,2:2,3:3,4:4,6:6
```

6. Good Time Intervals

http://cxc.harvard.edu/ciao/ahelp/times.html
http://cxc.harvard.edu/ciao/dictionary/gti.html
http://cxc.harvard.edu/ciao/ahelp/dmgti.html

- In a Chandra event file several 'times' are recorded both in the header via keywords [eg TSTART, TSTOP, EXPTIME etc.] and in one or several **Good Time Intervals** auxiliary files (the GTI blocks).

- GTIs are used to define **what times periods of the observation** can be used (i.e. **contain valid data**)

- The **dmgti** tool creates a Good Time Interval (GTI) filter file which represents the times of rows in the input file which pass the user-supplied constraints. As an example, the tool can be used to find out those times when the RATE column of a lightcurve lies below a set limit

7. Python: the scripting language in CIAO

- The scripting (or interpreted = no compilation is necessary) language supported in CIAO via Sherpa and ChIPS is Python (Python 3.5 as of CIAO 4.11)
- The software package is distributed with CIAO, but users may opt to run their own custom installation instead
- The CRATES and TRANSFORMS libraries provide access to the CIAO Data Model library and a high-level interface for performing World Coordinate System transformations.
- Access to CIAO's plotting and modeling packages is available through the ChIPS and Sherpa modules. Sherpa and ChIPS are importable modules for Python
- You DO NOT need to know Python to use Sherpa and Chips, but IF YOU DO, you will be able to use its capabilities in your analysis

Have fun using CIAO!