

# ANNOUNCEMENTS

The website for this workshop is

<https://cxc.cfa.harvard.edu/ciao/workshop/jan21/index.html>

Quick way to get there...

[cxc.harvard.edu](https://cxc.harvard.edu) → **Data analysis** → **Workshops**

You will find there

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

# An Introduction to CIAO

(Chandra Interactive Analysis of Observations)



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

## What is CIAO?

*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 more than 20 years!**

From: <http://cxc.harvard.edu/ciao/ahelp/ciao.html> or  
> **ahelp ciao** on the ciao command line

AHELP for CIAO 4.13

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 and Fitting

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.

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

- 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)

## What is CIAO? In practice

[cxc.harvard.edu/ciao](https://cxc.harvard.edu/ciao)

The screenshot shows the CIAO website homepage. At the top, there is a navigation bar with links for CXC HOME, PROPOSER, ARCHIVE, DATA ANALYSIS, INSTRUMENTS & CALIBRATION, and FOR THE PUBLIC. A search bar is also present. The main header reads "CHANDRA INTERACTIVE ANALYSIS OF OBSERVATIONS" with a tagline "from 's'ciavo', 'I am your servant' in Venetian dialect". Below this, a brief description states that CIAO is software developed by the Chandra X-Ray Center for analyzing data from the Chandra X-ray Telescope. A secondary navigation bar lists various Chandra instruments: Sherpa | DS9 | ChaRT | MARX | CALDB | CSC 2 | CSC 1.1 | TGcat.

The main content area is divided into several columns:

- Download CIAO/CALDB:** Offers to "Install CIAO 4.13 & CALDB 4.9.4 with conda" or "Install with ciao-install". It includes links for release notes, installation instructions, and frequently asked questions.
- What has changed?:** Announces a new release of CIAO and provides information on updates, version history, and release notes.
- Where should I begin?:** Provides resources for new users, including a primer, welcome page, quick start guide, and installation instructions.
- I need help!:** Offers help pages, frequently asked questions, and known bugs.
- I need more!:** Provides advanced resources like manuals, understanding the Chandra PSF, and galleries.
- Citing CIAO:** Provides information on how to cite the CIAO software in publications.

A sidebar on the left contains a comprehensive menu of links categorized by topic, such as Introduction, Download CIAO, Data Analysis, Documentation, SHERPA, Scripting in CIAO, Data Products, PSF CENTRAL, and WORKSHOPS.



## 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. As of CIAO 4.13 **prism** can be called via ds9.

# FITS Format

<http://heasarc.nasa.gov/docs/heasarc/fits.html>

•

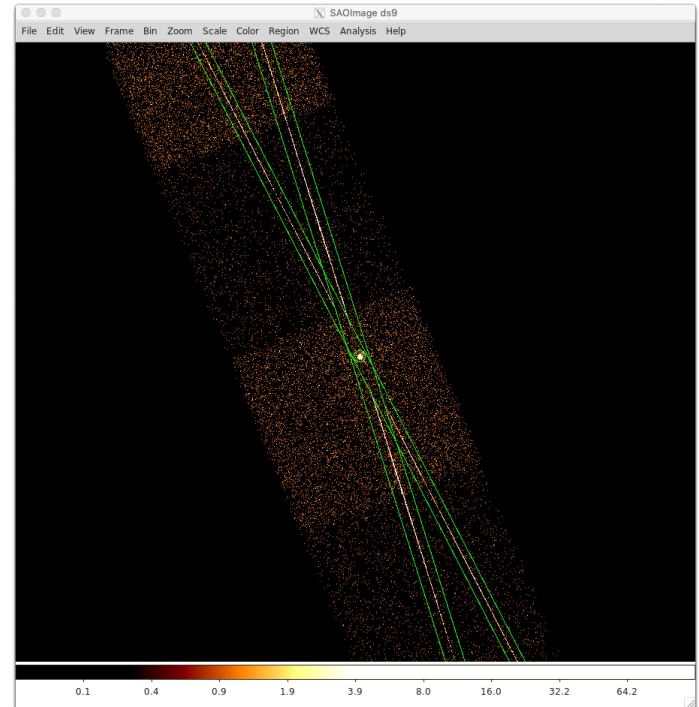
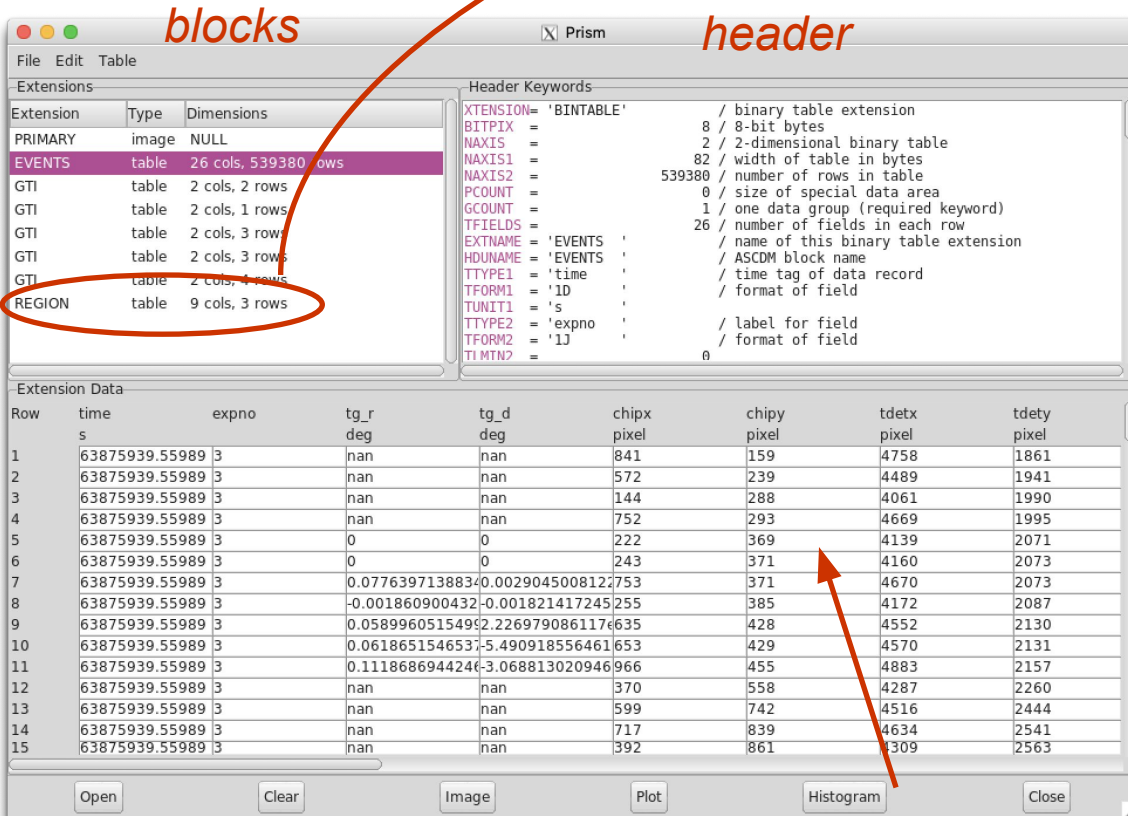
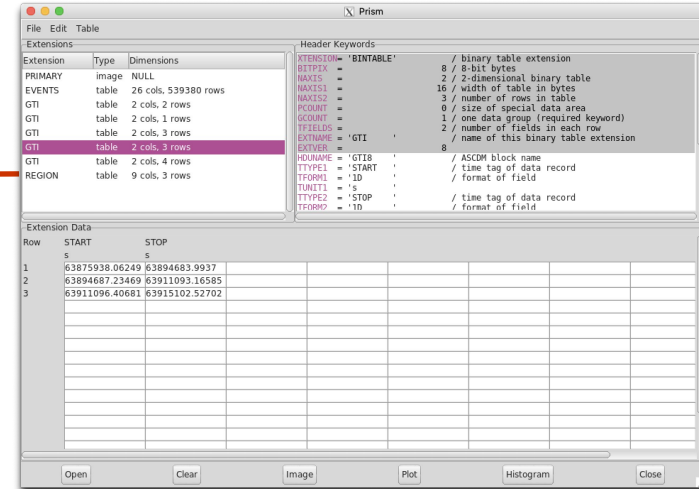
- Binary File Format that is “Self Documenting”
- Named Extensions, containing Headers and Data
- Extensions can be referred to by Name or Number
- Headers Describe Contents, Format, Processing History
- Keywords contain descriptive data
- Data are rows & columns containing values, arrays, images ...

**PRISM: examines FITS files, performs quick plots**

`ciao> prism acisf00459N004_evt2.fits`

Or  
`ciao> ds9 -prism acisf00459N004_evt2.fits`

Or  
`ds9` → File Open... → File Prism...



*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**)
- Parameters can be set via:
  - a number of routines (eg **punlearn**, **pset**, **plist**)
  - 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: dmcopy.par

```
ciao% 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)
```

```
ciao% 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.

```
ciao% pset dmcopy outfile=grating_reg2.fits
```

```
ciao% dmcopy
```

Input dataset/block specification (acisf00459N004\_evt2.fits.gz[REGION]):

Output dataset name (grating\_reg2.fits):

**Note the 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/dmbinring.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 FITS and ASCII datafiles.
- The Data Model 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.

## BENEFITS OF THE DATA MODEL

- 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
- **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” is represented by a filename followed by a series of optional qualifiers in square brackets [ ]:

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

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

the order of the qualifiers generally matters, however...

- not all qualifiers need to be present always

A few examples:

## Simple examples of “virtual files”

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

**acisf01843N001\_evt2.fits[EVENTS][time=84245787:84247000][cols #1,#2,#3]**

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

**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]  
[grade=0,2,3]

**columns/binning:**

[cols #1,#2,#3]

[cols time,ccd\_id,node\_id]

- 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`)

- An event file 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]

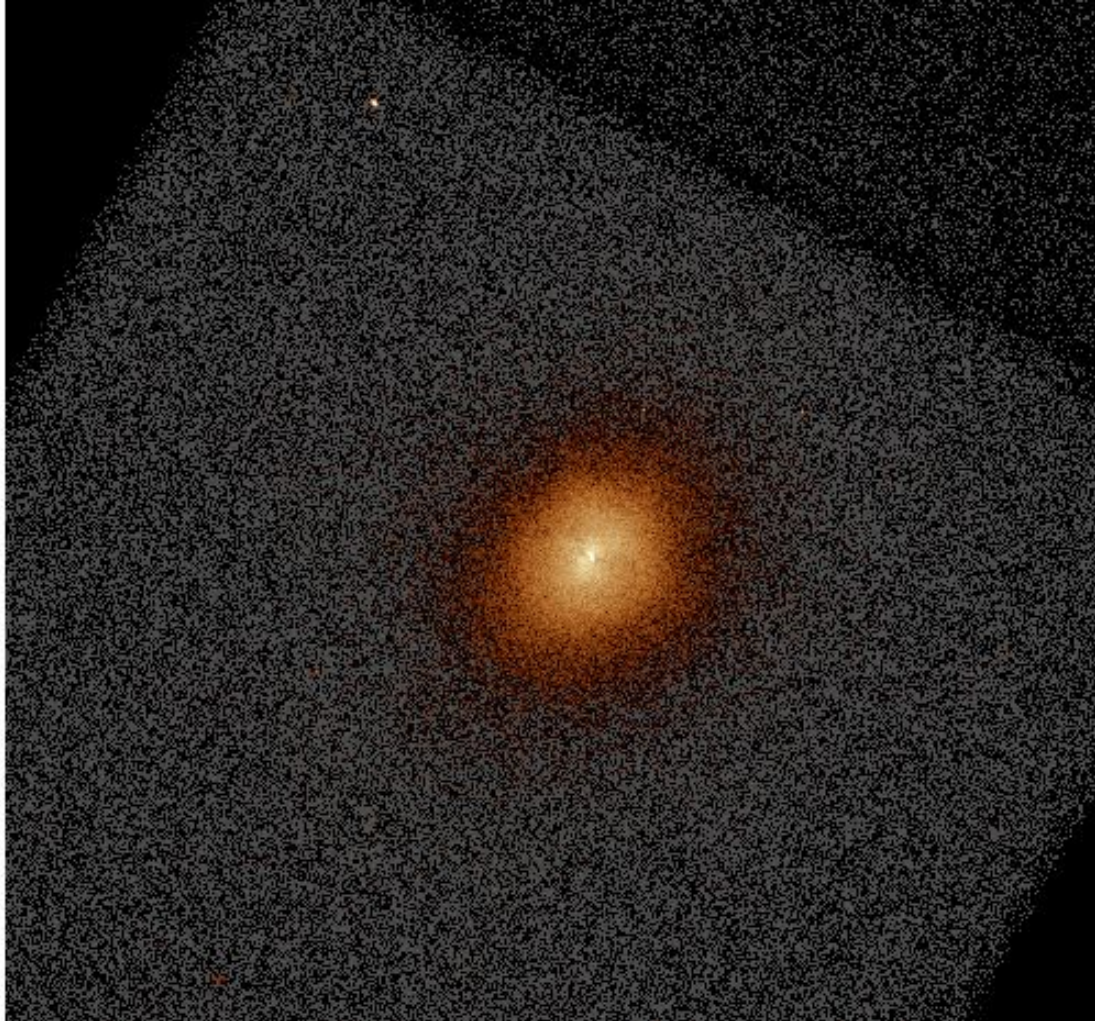
**filter:** [sky=region(mysource.reg)]  
[energy<7000]

**columns/binning:**

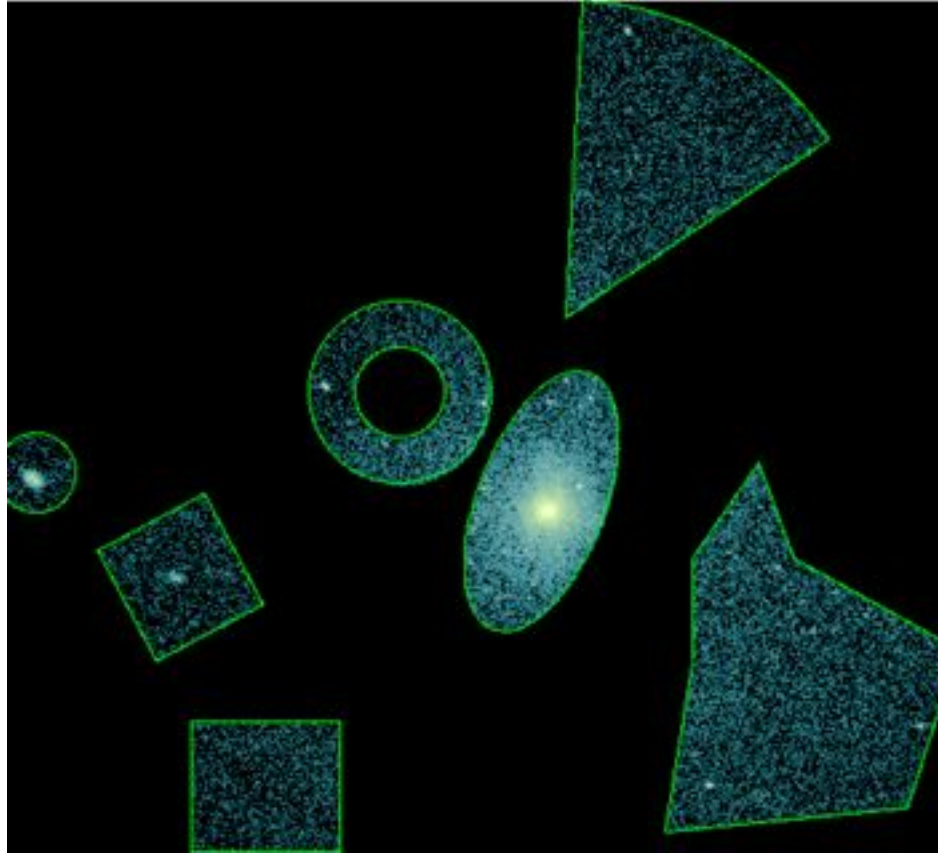
[bin pi=1:1024:1]

[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`



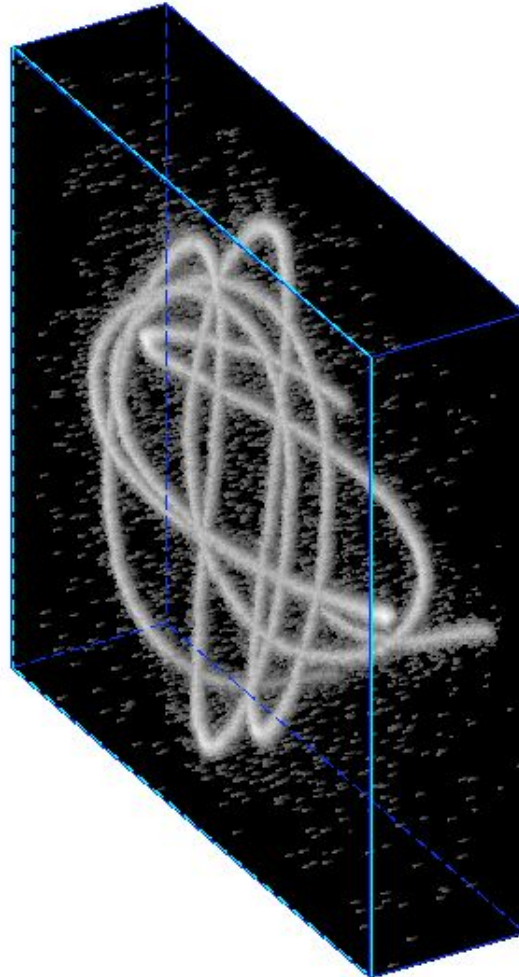
[dmcop](#) "ngc1404.img[sky=region(ngc1404\_sample.reg)]" ngc1404\_regfile.img



```
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>

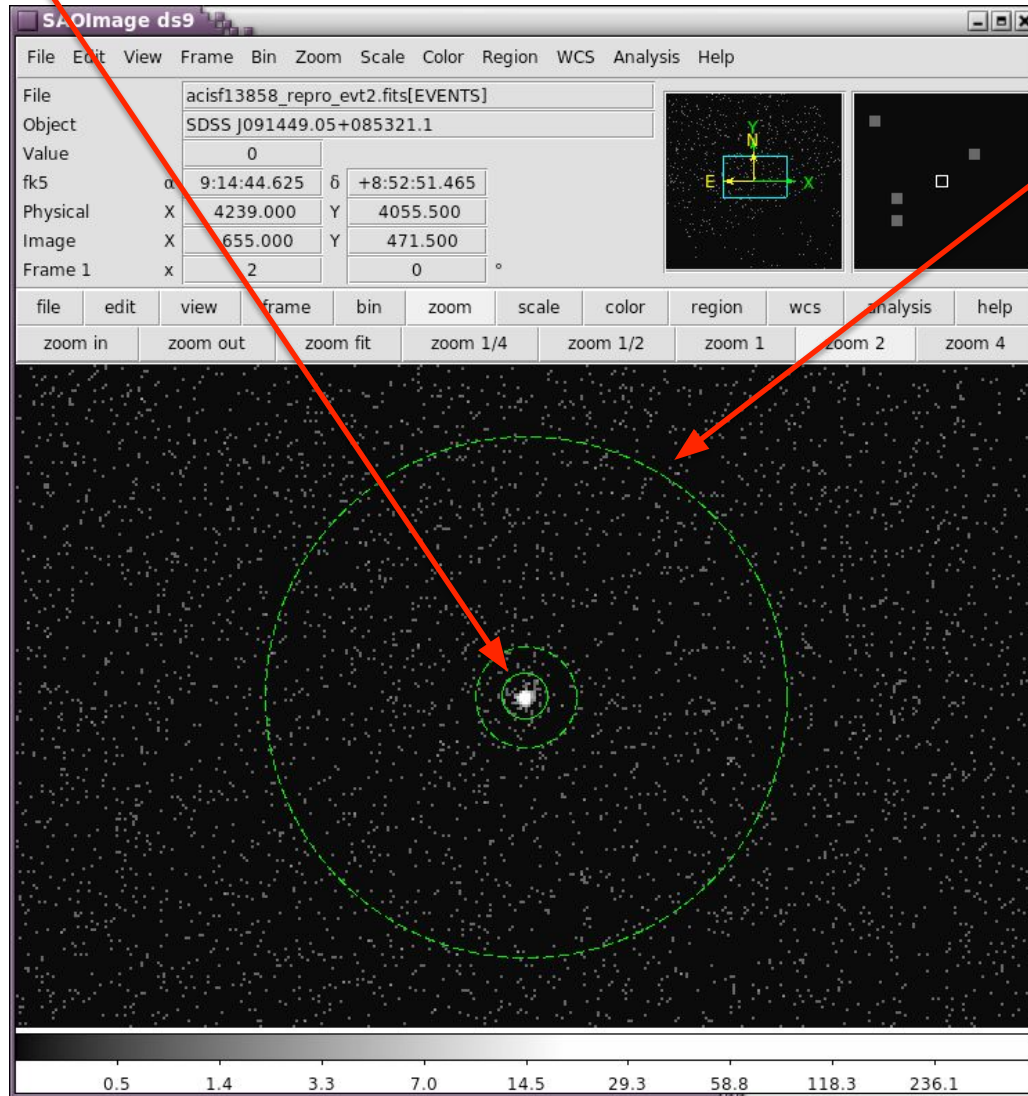
<http://cxc.harvard.edu/ciao/threads/regions/>

- 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.

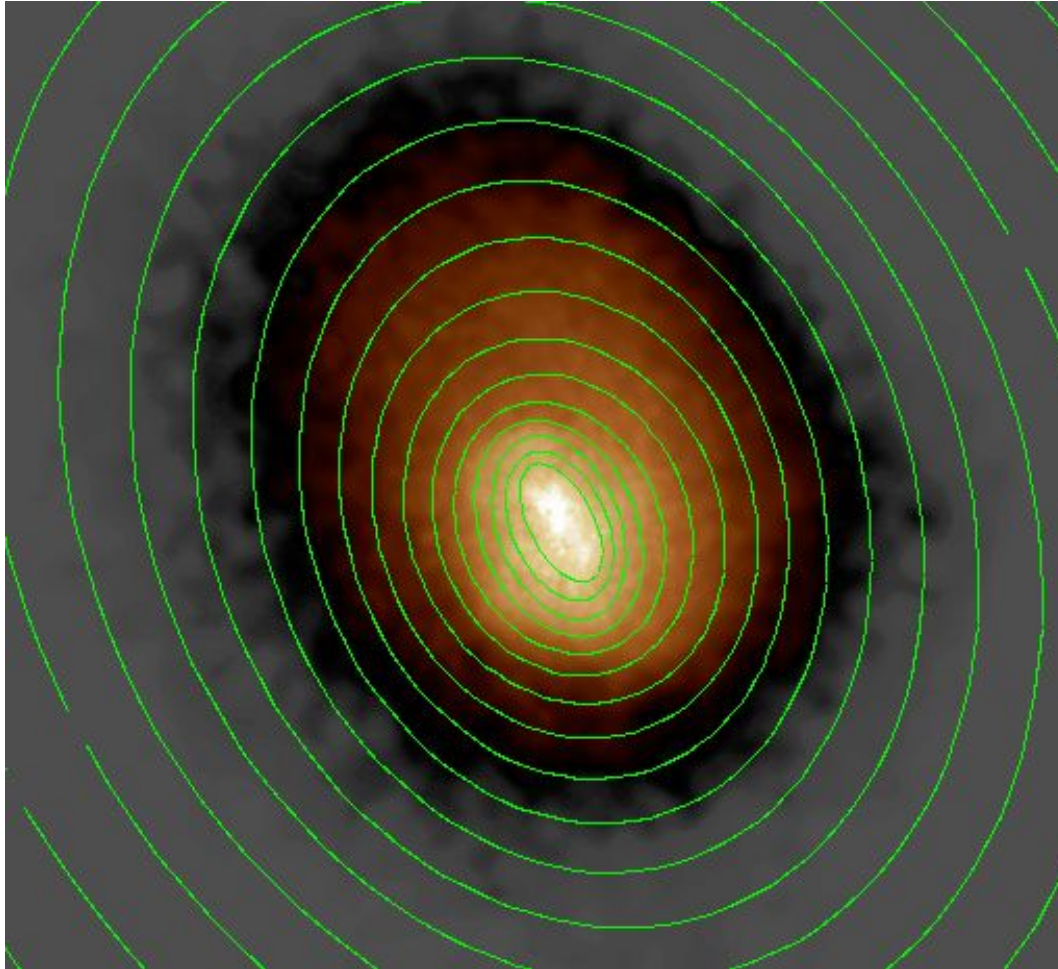


`circle(9:14:49.090,+8:53:21.231,4.083")`

`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>

- The subspace **records the filters applied to a file into the header**
- This allows subsequent tools to, for example, apply appropriate calibration information.
- **dmlist** can read the processing history using `opt=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 ---

<b>1 time</b>	<b>Real8</b>	<b>TABLE GTI3 444947637.1627430916:444970563.5344673395</b>
[...]		
<b>3 ccd_id</b>	<b>Int2</b>	<b>3:3</b>
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
<b>8 sky</b>	<b>Real4</b>	<b>Circle(4324,3676,50)</b>
<b>8 sky</b>	<b>Real4</b>	<b>Field area = 6.71089e+07 Region area = 7853.98</b>
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
<b>15 grade</b>	<b>Int2</b>	<b>0:0,2:2,3:3,4:4,6:6</b>

## 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 a user-supplied constraints.

## 7. Python: the scripting language in CIAO

- The scripting (or interpreted = no compilation is necessary) language supported in CIAO is Python (Python 3.7 as of CIAO 4.13). The CIAO Conda edition supports Python 3.8, 3.7 and 3.6
- 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 modeling package is available through the [Sherpa](#) module. Sherpa is an importable module for Python
- You DO NOT need to know Python to use Sherpa, but IF YOU DO, you will be able to use its capabilities in your analysis

## CIAO 4.12 - CONDA Installation

- As of version 4.12, CIAO can be installed using [conda](#)
- **conda** is a package management and environment management system which allows users to create multiple environments with multiple versions of packages which can co-exist on the same system.
- With conda, Linux and Mac users can choose to install CIAO-4.12 using either Python 3.7, 3.6, or 3.5. Using the conda package manager users can also install additional third-party packages.
- **We would like your feedback about the CIAO Conda installation: successes and problems. Send feedback to the CXC Helpdesk**

# Have fun using CIAO!

