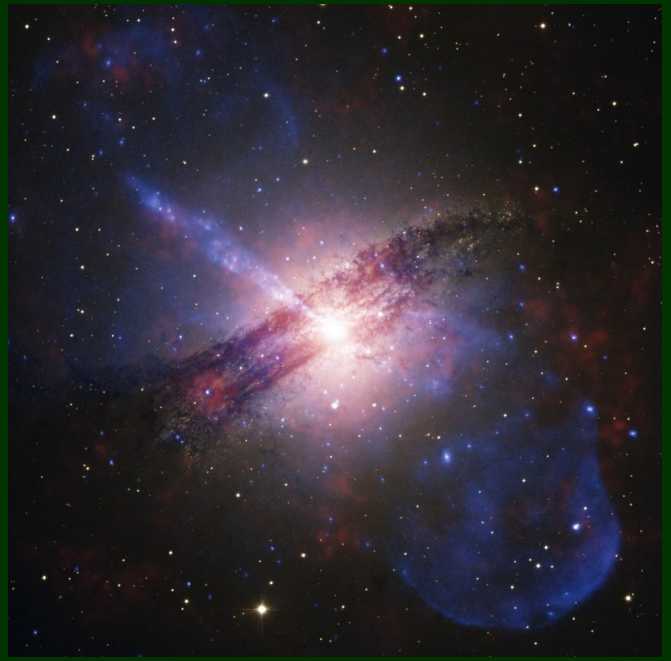
Introduction to CIAO

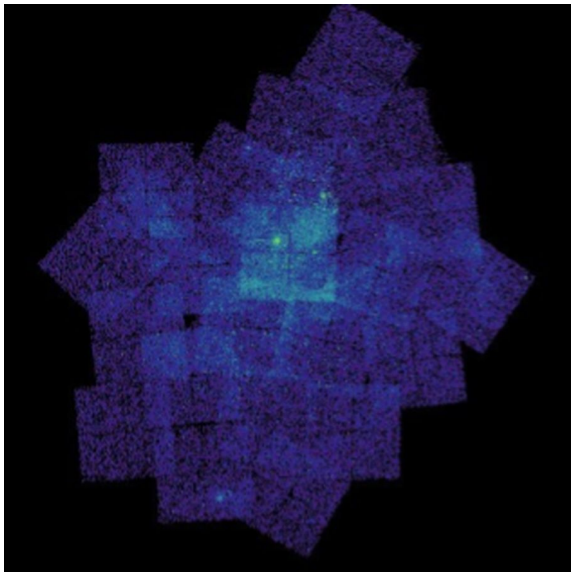
Chandra Interactive Analysis of Observations

Antonella Fruscione
Chandra X-ray Center, Center for Astrophysics | Harvard & Smithsonian
6th Advanced Arab School for Astrophysics



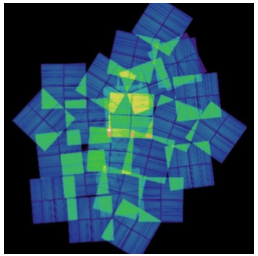
Centaurus A · Chandra Photo Album

From raw data to science: η Carinae



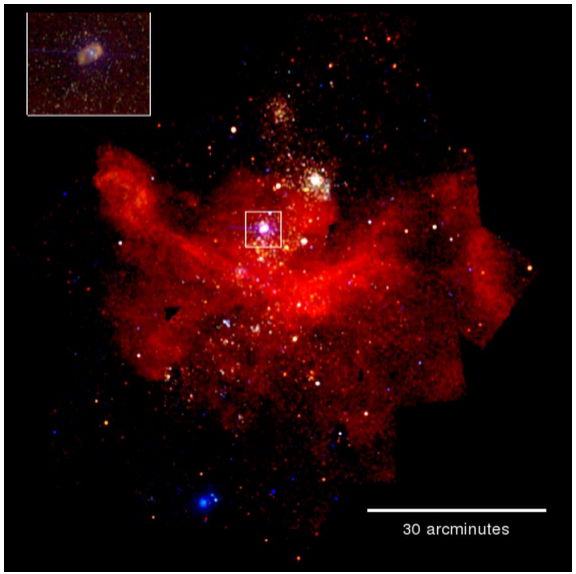
Before: raw counts
41 ACIS-I observations, 0.5–7 keV

CIAO



÷ exposure map

Fruscione et al. 2026, Fig. 7

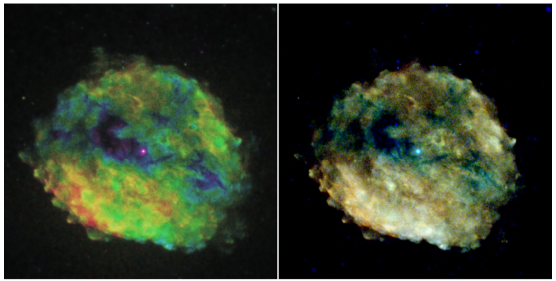


After: fluxed true color
R 0.5–1.2 · G 1.2–2 · B 2–7 keV

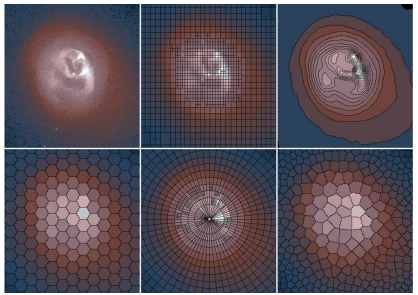
chandra_repro → merge_obs → true_color

What CIAO can do

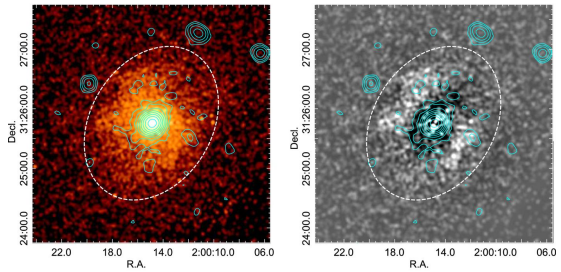
Figures: Fruscione et al. 2026, ApJ



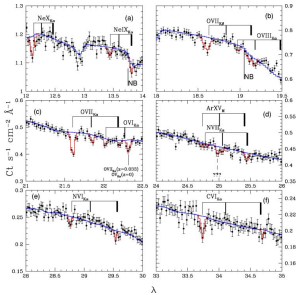
True-color images
RCW 103 · true-color thread



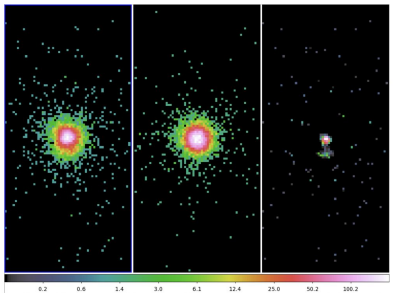
Adaptive binning
several algorithms · binning gallery



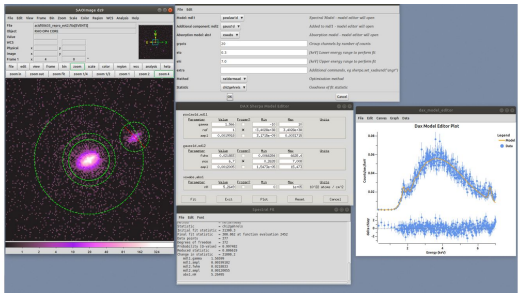
2-D image modeling
NGC 777 · Sherpa



High-resolution spectra
Mrk 421, LETG · grating threads



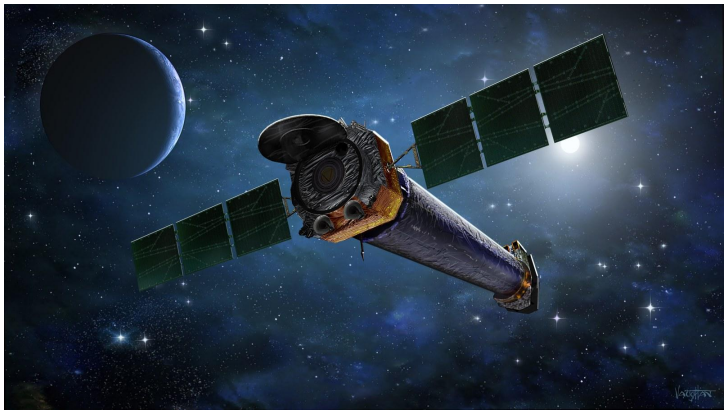
PSF simulation & deconvolution
T Canis Majoris · PSF tools



Analysis from DS9 menus
spectral fit with DAX · `dax`

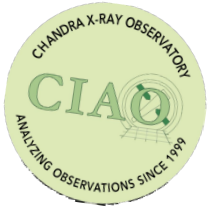
Where this talk fits in your day

1. **Chandra & X-ray astronomy**
   the telescope and the photons
2. **CIAO I: Introduction**
   concepts & tools ← you are here
3. **CIAO II: Imaging, spectra, CSC**
   doing science with CIAO
4. **SAOImageDS9**
   looking at the data
5. **Statistics in X-rays**
   counting photons properly
6. **Hands-on**
   afternoon exercises



Chandra X-ray Observatory (illustration) · Chandra Photo Album

What is CIAO?



The software package the Chandra X-ray Center provides to analyze Chandra data

- Command-line tools plus Python, for Linux and macOS
- Shares code with the Chandra standard processing pipeline
- Mission-independent design: also used for XMM-Newton, NuSTAR and multiwavelength data
- Easy for beginners, powerful for experts

"ciao" comes from the Venetian s'ciavo, "I am your servant"

The reference: Fruscione et al. 2026, ApJ

1999
CIAO 1.0 released

4.18
current version (Dec 2025)

130+
core tools, plus many scripts

1500+
installations per year

Chandra data are lists of events

Every event recorded by the detector is one row in the event file

time (s)	x	y	energy (eV)	ccd_id
84245787.6	4096.3	4102.8	1324.5	7
84245790.8	4021.7	3988.1	6402.0	7
84245794.0	4097.1	4101.9	873.2	7
84245797.2	3955.4	4210.6	2210.9	6
…	…	…	…	…

Illustrative rows. Real files carry ~20 columns: chip & detector coordinates, PI, grade, status…



Bin x, y → image



Bin energy → spectrum



Bin time → light curve

CIAO keeps every event and lets you choose how to look at them

A photon is not an event

Photon
what arrives at the telescope

- true energy E
- true position on the sky
- exact arrival time

mirror + detector

Event
what ACIS records

- PHA ⟶ PI: a measured channel
- chip pixel ⟶ sky x, y (aspect)
- time of the 3.2 s CCD frame
- grade (charge pattern) and status flags

Photons lost
effective area, chip gaps, bad pixels

Events with no photon
particle background, removed by grade & status filters

Energy smeared
PI is not E: the RMF links them

Position blurred
PSF and aspect: x, y is not the true position

2 photons ⟶ 1 event
pile-up in bright sources

An event file is a list of events. The calibration (ARF, RMF, exposure maps, PSF) is how CIAO gets back from events to photons. Dictionary: event

What's in the CIAO box

Command-line tools

dmcopy, dmlist, dmextract, mkarf… small tools that do one job

Analysis scripts

chandra_repro, fluximage, specextract, srcflux: whole threads in one command

Sherpa

modeling and fitting of spectra, images and light curves

SAOImageDS9 + dax

display, regions, and CIAO analysis from DS9 menus

Python libraries

crates (files), transforms (coordinates), Matplotlib, Jupyter

CALDB

files that describe how Chandra's mirrors and detectors behave; CIAO finds them for you

Installing CIAO 4.18

ciao-install

new to conda? start here

conda

if you already use conda environments

macOS installer

.dmg with everything, incl. CALDB

```
# 1) ciao-install: download the script from the CIAO download page, then
% bash ciao-install
# 2) or conda
% conda create -n ciao-4.18 -c https://cxc.cfa.harvard.edu/conda/ciao -c conda-forge \
    ciao sherpa ds9 ciao-contrib caldb_main marx numpy=2.3.5
% conda activate ciao-4.18
% ciaover -v          # check versions of CIAO, contrib and CALDB
```

Follow the current instructions at cxc.cfa.harvard.edu/ciao/download. The NumPy pin is there because CIAO 4.18 does not support NumPy 2.4+. Then try the CIAO startup thread.

A typical Chandra analysis



Download	Reprocess	Clean & filter	Extract	Model & fit
`download_chandra_obsid`	`chandra_repro`	`deflare`, `dmcopy`	`fluximage`, `specextract`, `dmextract`	`Sherpa`

Every step has a CIAO thread: a step-by-step recipe at cxc.cfa.harvard.edu/ciao/threads. Read it line by line the first time, then reuse it.

Steps 1–2: get the data, then reprocess

```
% download_chandra_obsid 1843
# creates 1843/primary and 1843/secondary

% chandra_repro indir=1843 outdir=1843/repro
# applies the latest calibration (CALDB)
# → 1843/repro/acisf01843_repro_evt2.fits
```

Why reprocess?
Archive files were made with the calibration available at the time. chandra_repro gives you files consistent with the CALDB you have installed today.
Threads: download data · reprocess · data products

Key files

evt2 level 2 event file: the events

asol1 aspect solution: where we pointed

bpix1 bad pixels

flt1 good time intervals

fov1 field-of-view regions

msk1 valid detector area

PART 2

Seven ideas behind CIAO

File format · Parameter files · Data Model filtering & binning · Regions · Subspace · Good Time Intervals · Python



Crab Nebula · Chandra Photo Album

1 · FITS files: headers and data blocks

`acisf01843_repro_evt2.fits`

PRIMARY	header only
EVENTS	table: one row per event
GTI	table: good time intervals (one per chip)

Each block = header (processing history, times, calibration) + data. *ahelp dmlist*

```
% dmlist evt2.fits blocks
% dmlist "evt2.fits[events]" cols
% dmlist "evt2.fits[events]" header
% dmlist "evt2.fits[events][#row=1:5]" data

# or browse graphically:
% ds9 -prism evt2.fits
```

2 · Parameter files remember your settings

```
% plist dmcopy

 Parameters for /home/you/cxcds_param4/dmcopy.par
       infile =                  Input dataset/block specification
      outfile =                  Output dataset/block specification
      (kernel = default)         Output file format type
      (option = )                Option
     (verbose = 0)               Debug Level(0-5)
     (clobber = no)              Clobber existing file
```

`punlearn dmcopy`
reset to defaults: do this first!

`pget dmcopy outfile`
read a parameter

`pset dmcopy clobber=yes`
set a parameter

`dmcopy in.fits out.fits clob+`
or just give everything on the command line

3 · The Data Model: one syntax for every file

```
evt2.fits[events][energy=500:7000][bin x=::4,y=::4]
```

file
any FITS (or ASCII) file

block
which extension

filter
keep only rows that match

binning / columns
histogram into an image, or select columns



Virtual files. Any tool accepts a filtered or binned file as input, with no intermediate file on disk. The same syntax works for events, images and tables, and for any column.
ahelp: dm · dmfiltering · dmbinning · dmregions

Virtual files: filter and bin on the fly

```
evt2.fits[events][time=84245787:84247000][cols time,energy]
```
a time slice, keeping two columns

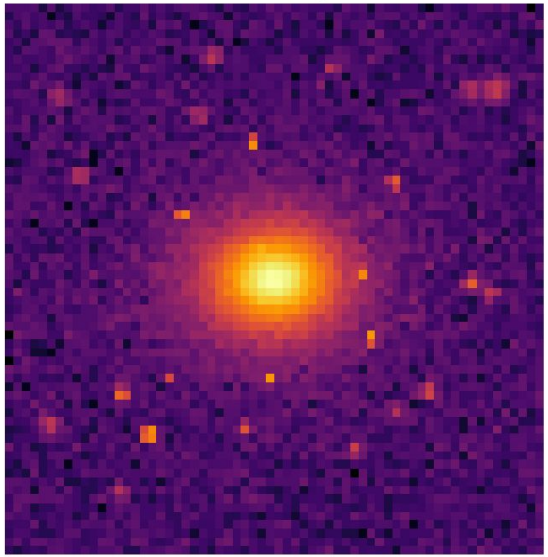
```
evt2.fits[events][grade=0,2,3,4,6][ccd_id=7]
```
standard ACIS grades on chip S3 only

```
evt2.fits[sky=region(src.reg)][bin pi=1:1024:1]
```
a spectrum of the source region (typical dmextract input)

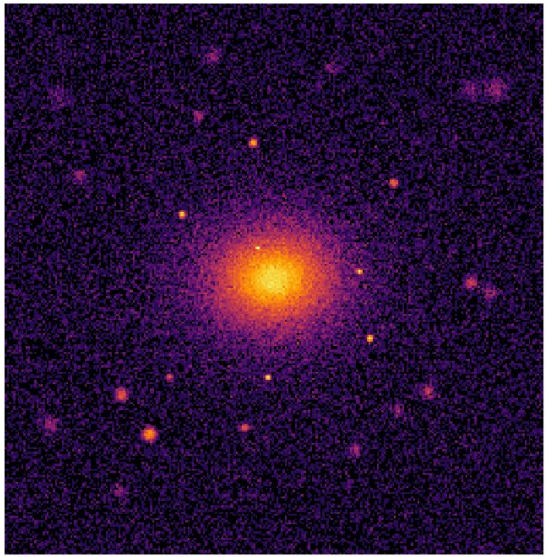
```
evt2.fits[energy=500:7000][bin x=3500:4500:2,y=3500:4500:2]
```
a 0.5–7 keV image, 2-pixel bins (typical dmcopy input)

dmcopy: same events, different pixels

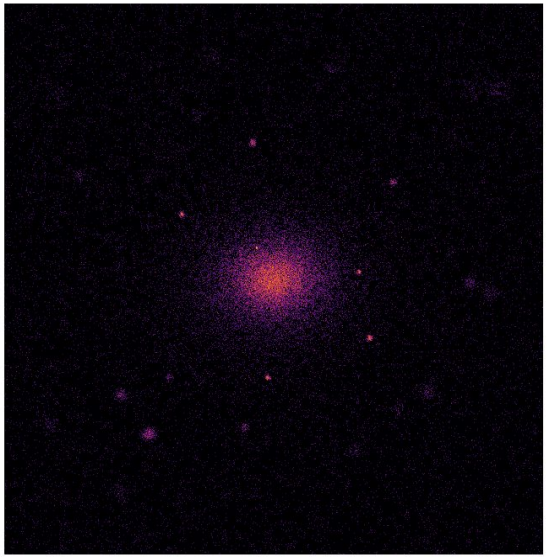
[bin x=::16,y=::16]



[bin x=::4,y=::4]



[bin x=::1,y=::1]



```
% dmcopy "evt2.fits[energy=500:7000][bin x=::4,y=::4]" img_bin4.fits
```

Four tools you will use every day

dmlist

look inside: blocks, columns, header, data

```
dmlist evt2.fits blocks
```

dmcopy

filter and bin; write the result to a file

```
dmcopy "evt2.fits[ccd_id=7]" s3.fits
```

dmextract

make spectra, light curves, radial profiles

```
dmextract "evt2.fits[bin time=::200]"
   lc.fits opt=ltc1
```

dmstat

statistics of images and columns

```
dmstat "evt2.fits[cols energy]"
```

4 · Regions select where

```
# src.reg  (draw in DS9, save in ciao format)
circle(9:14:49.090,+8:53:21.231,4.083")

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

Shapes
circle, ellipse, box, polygon, annulus, pie…

Combine
“+” (or) for union, “*” (and) for intersection, “−” to exclude

```
% dmcopy "evt2.fits[sky=region(src.reg)]" src.fits
% dmcopy "evt2.fits[exclude sky=region(pts.reg)]" diffuse.fits
```

Units
sky pixels or celestial coordinates

Regions are spatial filters: the same files work in DS9, in dmcopy, and in specextract

5 · Subspace and 6 · Good Time Intervals

Subspace: the file remembers its filters

Every filter you apply is recorded in the header, so later tools apply the right calibration (e.g. only the chips and region you kept).

```
% dmlist "evt2.fits[ccd_id=3,
    sky=circle(4324,3676,50)]" subspace
```

GTIs: when the data are good

GTI blocks list the time ranges used for science; exposure time comes from them.

Remove background flares → new GTIs → correct exposure.

```
% deflare lc.fits clean.gti
method=clean
% dmcopy "evt2.fits[@clean.gti]" \
    clean.fits
```

7 · Python under the hood

```
from pycrates import read_file
import matplotlib.pyplot as plt

cr = read_file("evt2.fits[energy=500:7000]")
energy = cr.get_column("energy").values

plt.hist(energy / 1000, bins=100)
plt.xlabel("Energy (keV)")
plt.show()
```

crates
read & write files, with Data Model filters

transforms
coordinates: sky ↔ chip ↔ RA/Dec

sherpa
fitting, as a Python module

Matplotlib & Jupyter
plots and notebooks

CIAO 4.18 ships Python 3.12. You do not need Python to use CIAO, but it makes scripting and reproducible analysis easy.

Where to get help

ahelp
ahelp dmcopy · ahelp -k spectrum

Threads
step-by-step recipes for every analysis

FAQ & known bugs
check here when something looks odd

Dictionary & guides
terms (PI, ARF, GTI…), PSF and calibration guides

CXC Helpdesk
real people answer: cxc.harvard.edu/help

Videos
CIAO demos on YouTube (4ciaodemos)

All at cxc.cfa.harvard.edu/ciao

TAKE-HOME

Events in, science out

- Chandra data are event lists: you choose how to bin them
- Learn the Data Model syntax: it works with every tool
- punlearn, read the thread, and use ahelp
- Next: imaging, spectra and the Chandra Source Catalog



M82 · Chandra Photo Album