CIAO Quick Start

Abbreviated CIAO threads to get started with the most common Chandra data reduction tasks.

I want to ...

… install CIAO

Goto http://cxc.cfa.harvard.edu/ciao/download

Select **Standard Install with the base CALDB**

Save `ciao-install`

```
$ bash ~/Download/ciao-install
```

… setup for CIAO

```
$ alias ciao=”source /soft/ciao/bin/ciao.bash”

$ ciao
```

We recommend that users add the `ciao` alias to their `~/.bashrc` file.

… get help about a CIAO command or concept

```
$ ahelp dmlist

$ ahelp regions
```

CIAO includes over 1200 help files covering all the Sherpa and CHIPS commands, individual tools & scripts, and Chandra & CIAO concepts.

… find observations of my source

```
$ find_chandra_obsid 3c273
```

`find_chandra_obsid` *only works for public, non-proprietary, data. It can also search by position and radius, filter by instrument and grating, and download data.*

… download data

```
$ download_chandra_obsid 4425

$ chandra_repro 4425 outdir=
```

We encourage users to apply the most recent calibrations to their datasets when the data are retrieved from the archive.

… get information about a file (column, keywords, blocks)

```
$ dmlist 635/repro/acisf00635_repro_evt2.fits header,clean
```

`dmlist` *can also display information about coordinate systems and any filters ("subspace"). It works with images and tables and mutli-extension FITS or ASCII files.*

`prism` *can also be used to display the contents of a file.*

… extract spectrum for a point-like source

```
$ ds9 4425/repro/acisf04425_repro_evt2.fits
```

Create a region and save as `src.reg`

```
$ specextract \
 "4425/repro/acisf04425_repro_evt2.fits[sky=region(src.reg)]" \
 mysrc
```

`specextract` *has options for including background, extended sources, PSF corrections, grouping, and combining datasets.*

… extract high-resolution grating spectrum

```
$ tgextract acisf13721_repro_evt2.fits my_src.pha2
```

```
$ mktgresp my_src.pha2 acisf13721_repro_evt2.fits my_src
```

Grating data retrieved from the Chandra archive already includes the PHA file for the brightest source in the field. Additionally, data run through `chandra_repro` *already has the response files created. As does data retrieved from TGCat*

Users can use `tg_create_mask`, `tgextract`, *and* `tgextract2` *to customize the spectrum.*

… combine spectra

```
$ combine_spectra "CXOJ123538.4+621643/*/*/*pha3.fits.gz" cxo
```

Users are not required to reproject the data to the same point nor must they apply fine astrometric corrections before combining spectra.

```
$ combine_grating_spectra tgcat/obs_5422_tgid_3921/pha2 \
   outroot=obs5422 add_plusminus=yes \
   arf="tgcat/obs_5422_tgid_3921/*.arf" \
   rmf="tgcat/obs_5422_tgid_3921/*.rmf"
```

`combine_grating_spectra` *can combine data from multiple observations in addition to combining positive and negative orders.*

… convert count rate to flux

```
$ modelflux mysrc.arf mysrc.rmf \
  model=xspowerlaw.p1 paramval="p1.PhoIndex=2.0" \
  absmodel=xsphabs.a1 absparams="a1.nH=0.02" \
  emin=0.5 emax=7.0 rate=0.1
```

`modelflux` *has access to all* `sherpa` *models and can be used with arbitrary energy ranges and model parameters.*

Users are discouraged from using `PIMMS` *to estimate the observed flux.*

… estimate the flux for a point-like source

```
$ srcflux 4425/repro/acisf04425_repro_evt2.fits \
  "11:33:05,+25:53:55.4" mysrc
```

`srcflux` *has options for PSF corrections, energy bands, confidence intervals (including upper-limits), spectral models, and user supplied regions.*

… make exposure corrected images

```
$ fluximage 635/repro/acisf00635_repro_evt2.fits mydata
```

`fluximage` *has options for energy bands, binsize (resolution), and for HRC background subtraction.*

… create 3 color (RGB) image

```
$ fluximage 214/repro/acisf00214_repro_evt2.fits cas_a \
    bands="soft,medium,hard" bin=4

$ ds9 -rgb -red cas_a_soft_flux.img -green \
  cas_a_medium_flux.img -blue cas_a_hard_flux.img
```

There are multiple ways to perform tasks in `ds9`*, including through the GUI, on the command line, via XPA or SAMP, and even through HTML.*

… compute source centroid

```
$ dmstat \
  "acisf04425_repro_evt2.fits[sky=region(src.reg)][bin sky=1]"
```

`dmstat` *compute some basic statistics (min, max, mean, median, standard deviation) for columns in a table, or for image pixel values.*

… detect sources

```
$ fluximage "635/repro/acisf00635_repro_evt2.fits[ccd_id=0:3]"\
  acis_I_rOph binsize=1 clob+

$ mkpsfmap acis_I_rOph_broad_thresh.img \
  acis_I_rOph_broad.psfmap energy=2.3 ecf=0.9 spectrum="" clob+

$ wavdetect infile=acis_I_rOph_broad_thresh.img \
  psffile=acis_I_rOph_broad.psfmap \
  expfile=acis_I_rOph_broad_thresh.expmap \
  scales="1.4 2 4 6 8 12 16 24 32" outfile=acis_I_rOph.src \
  scellfile=acis_I_rOph.cell imagefile=acis_I_rOph.recon \
  defnbkg=acis_I_rOph.nbkg
```

`wavdetect` *has options for setting significance threshold (false source rate) and user supplied background*

`wavdetect` *is* ***not*** *a photometric tool. While it does provide an estimate of the counts, we* ***strongly*** *recommend users compute flux separately, eg using* `srcflux`

```
$ srcflux 635/repro/acisf00635_repro_evt2.fits \
  pos=acis_I_rOph.src outroot=acis_I_rOph
```

… extract light curve (check for variability)

```
$ dmextract
\
"acisf00635_repro_evt2.fits[energy=500:7000][sky=ellipse(4053,3
210,16,10,38)][bin time=::3000]" flare.lc op=ltc1 clob+
```

or

```
$ glvary
\
"acisf00635_repro_evt2.fits[energy=500:7000][sky=region(ciao.re
g)]” vary.prob vary.lc none clob+
```

Both `dmextract` *and* `glvary` *make use of Good Time Intervals and an optional time-dependent efficiency factors file.*

… make a radial profile

```
$ dmextract hrcf17395_repro_evt2.fits"[bin
sky=annulus(16635,15690,0:3200:100)]" \
coma_prof.rad exp=coma_wide_thresh.expmap clob+ op=generic
```

`dmextract` *can also extract the radial profile from an image rather than an event file. It can also extract the background and provide net quantities.*

… search Chandra Source Catalog

```
$ search_csc 3c273 radius=1 outfile=coma.tsv
```

or

```
$ obsid_search_csc 635 outfile=rho_oph_635.tsv download=all \
  file=pha,arf,rmf
```

`search_csc` *and* `obsid_search_csc` *can retrieve all the columns in the CSC along with all the files and any the limiting sensitivity in any of the valid energy bands.*

… remove background flares

```
$ dmextract “acisf02233_repro_evt2.fits[energy=500:7000][bin
time=::254]” flare.lc op=ltc1

$ deflare flare.lc flare.gti clean plot+

$ dmcopy “acisf02233_repro_evt2.fits[@flare.gti]” clean_evt2.fits
```

`deflare` *has different flare cleaning options and parameters that can be adjusted to optimize flare rejection.*

Users should carefully consider whether the time lost for their source is more important than the increased background during the flare.

… register (ie align) fields

```
$ wcs_match acisf02423_broad.src acisf02233_broad.src \
  cdfn_match.out  wcsfile=acisf02423_broad_thresh.img \
  method=trans radius=2

$ wcs_update 2423/repro/pcadf099299289N003_asol1.fits \
   2423/repro/updated_asol1.fits cdfn_match.out \
   wcs=acisf02423_broad_thresh.img

$ wcs_update 2423/repro/acisf02423_repro_evt2.fits  \
  none cdfn_match.out wcs=acisf02423_broad_thresh.img

$ dmhedit 2423/repro/acisf02423_repro_evt2.fits op=add \
  key=ASOLFILE value=updated_asol1.fits file=none
```

`wcs_match` *has several options to control the quality of the cross match. The* `dmhedit` *command is necessary to keep the updated output files in sync.*

… create mosaic

```
$ merge_obs acisf02233_repro_evt2.fits,acisf02423_repro_evt2.fits\
 cdfn

$ ds9 cdfn_merged_evt.fits
```

`merge_obs` *also creates combined images and exposure maps. Users can also select their own tangent point location and binning.*

… smooth image for publication

```
$ csmooth "acis_I_rOph_broad_thresh.img[bin sky=4]" none \
  rOph.asm mode=h verb=2 clob+ sigmin=3 sigmax=5 sclmax=50
```

`csmooth` *creates an adaptively smoothed, highly processed image. While not appropriate for photometric analysis it is very useful to help identify features in their images.*

… plot data

```
$ chips

chips> add_curve("mysrc.pi[cols channel,counts]")

chips> lc = read_file("flare.lc")
chips> t_start = lc.get_column("time_min").values
chips> t_stop = lc.get_column("time_max").values
chips> rate = lc.get_column("count_rate").values
chips> add_histogram( t_start, t_stop, rate )

chips> add_image("rOph.asm")
```

`chips` *supports line & scatter plots, histograms, contours and 2D images.*

… model spectra (and images, and generic datasets)

```
$ sherpa

sherpa> load_data("mysrc.pi")
sherpa> set_source(xsphabs.a1*xspowerlaw.p1)
sherpa> a1.nH=0.1
sherpa> freeze(a1)
sherpa> fit()
sherpa> plot_fit()
sherpa> conf()
```

`sherpa` *is a general modeling and fitting application. It supports 1D and 2D datasets, various optimization methods, and produces confidence limits.*

… do something else?

CIAO has over 200 tools and scripts!

CIAO supports specialized Chandra analysis such as Solar System Objects and Optical Monitor data as well as generic multi-mission & multi-wavelength analysis.