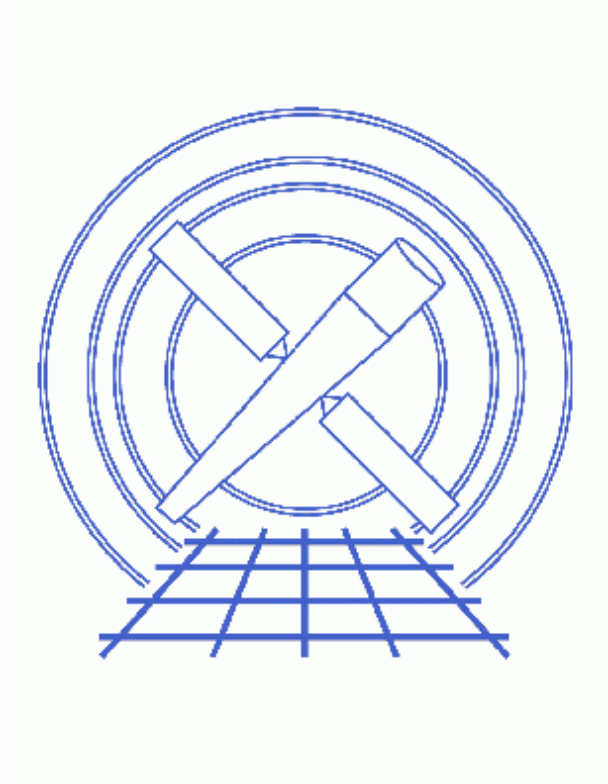
HETG/ACIS-S Grating ARFs



CIAO 4.1 Science Threads

Table of Contents

URL: http://cxc.harvard.edu/ciao/threads/mkgarf_hetgacis/ Last modified: 16 June 2009

HETG/ACIS-S Grating ARFs

CIAO 4.1 Science Threads

Overview

Last Update: 16 Jun 2009 - added About the Chandra Grating Data Archive and Catalog section

Synopsis:

`fullgarf` is a script that creates a grating ARF for a particular order and grating of an observation. While the `mkgarf` tool will create a grating ARF for an individual chip given an aspect histogram, this script creates ARFS for each chip, creating aspect histograms as necessary. The script then combines the individual ARFS into one for the full array.

Purpose:

To create grating ARFs for an HETG/ACIS-S observation.

Read this thread if:

you are working with an HETG/ACIS-S observation and intend to model and fit the spectra.

Related Links:

- Analysis Guide for Chandra High Resolution Spectroscopy: an in-depth discussion of grating analysis.
- Continuous Clocking Mode why topic: additional information for users working with CC-mode data.
- ACIS QE Degradation why topic: correcting for the change in low-energy ACIS QE associated with the deposition of one or more materials on the ACIS detectors or optical blocking filters.

Proceed to the HTML or hardcopy (PDF: A4 | letter) version of the thread.

About the Chandra Grating Data Archive and Catalog

The Chandra Grating Data Archive and Catalog (TGCat) is a browsable interface to analysis-quality spectral products (binned spectra and corresponding response files). TGCat makes it easy to find observations of a particular object, type of object, or type of observation, to quickly assess the quality and potential usefulness of the spectra with pre-computed graphics or custom-generated plots of binned and combined counts or fluxe spectra. Spectra, responses, event files, and summary products may be downloaded as a package.

TGCat runs standard CIAO tools, but also includes customized extractions for non-standard cases to refine the zeroth order position or to use regions appropriate for extended sources. Non-standard extractions details are provided in "verification and validation" comments for users.

Most public grating observations are available and new ones are added soon after they are released. See the list of of observations not included for exceptions. Many of the observations currently in this list will be included when we add enhanced processing for more difficult cases (multiple sources, very extended sources).

Please consider using the spectrum and responses (PHA, ARF, and RMF files) from TGCat in your analysis.

Get Started

Sample ObsID used: 459 (HETG/ACIS-S, 3C 273)

File types needed: evt2; pha2; asol1; bpix1; msk1; pbk0

If you created a new bad pixel file by running the New ACIS Bad Pixel File: Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread, use that file in this analysis. Otherwise, use the `bpix1.fits` file from the Archive. Make sure that you have set up ardlib to use the same bad pixel file.

It is assumed that you have created a gRMF for your observation by running the Create Grating RMFs for ACIS-S Observations thread, which is called `rmf.fits`.

ACIS CTI_APP Keyword Required

CIAO 4.1 and CALDB 4.1 require that ACIS event files have a `CTI_APP` header keyword to indicate whether the CTI correction has been applied. The older `CTI_CORR` keyword is no longer used.

To check for `CTI_APP`:

```
unix% dmkeypar input.fits CTI_APP echo+
# dmkeypar (CIAO 4.1): ERROR: Keyword 'CTI_APP' was not found in file 'input.fits'.
```

If `CTI_APP` is not found, follow the instructions in the ACIS CTI_APP Keyword Required section of the ACIS CTI Correction why topic to add the keyword before continuing.

This thread may produce incorrect results **without issuing an error** if the keyword is missing.

Download the script

This thread uses the `fullgarf` script; for information about the script, consult the help file ("`ahelp fullgarf`"). The script is part of the CIAO Scripts distribution. The CIAO scripts package should be the following version or newer:

```
unix% cat $ASCDS_CONTRIB/VERSION.CIAO_scripts
01 Jun 2009
```

Please check that you have at least this version of the scripts package installed before continuing. If you do not have the scripts installed or need to update to a newer version, refer to the Scripts page.

Using the mask file

To make sure that the gARF file is correct, set the `maskfile` parameter to the observation-specific msk1.fits file. **If you are working with continuous-clocking (CC) mode data**, be sure to read the ACIS Pipeline-Processed Mask Files caveat as well.

The ACIS Dead Area Correction

As of CIAO 4.0, the application of the dead area correction is **turned on** by default. The `pbkfile` and `dafile` parameters were added to `fullgarf` in order to apply the calibration.

The "dead area" is a fractional area loss per unit time due to cosmic ray flux incident on the ACIS detector. Calibration to account for this ACIS "dead area" was included in CALDB 3.3.0 on 15 December 2006. The correction is non-zero for the 8 front-illuminated (FI) ACIS chips; the effect is not detectable for the back-illuminated (BI) chips, so there is no correction applied to them.

Refer to the ACIS Dead Area Correction why topic for more information, including how to "turn off" the correction, e.g. if you would like to compare results with and without it applied.

Determine the Orders of the Observation

An ARF needs to be calculated for each order in the observation. We can use ***Prism*** to examine the pha2 file and determine how many orders there are:

```
unix% prism acisf00459N002_pha2.fits &
```

The `tg_m` column indicates the order of the observation (+/- 1, +/- 2, +/- 3) and the `tg_part` column indicates the grating (1 = HEG, 2 = MEG, 3 = LEG). The twelve rows (all +/- orders for HEG and MEG) for the HETG observation are shown in Figure 1.

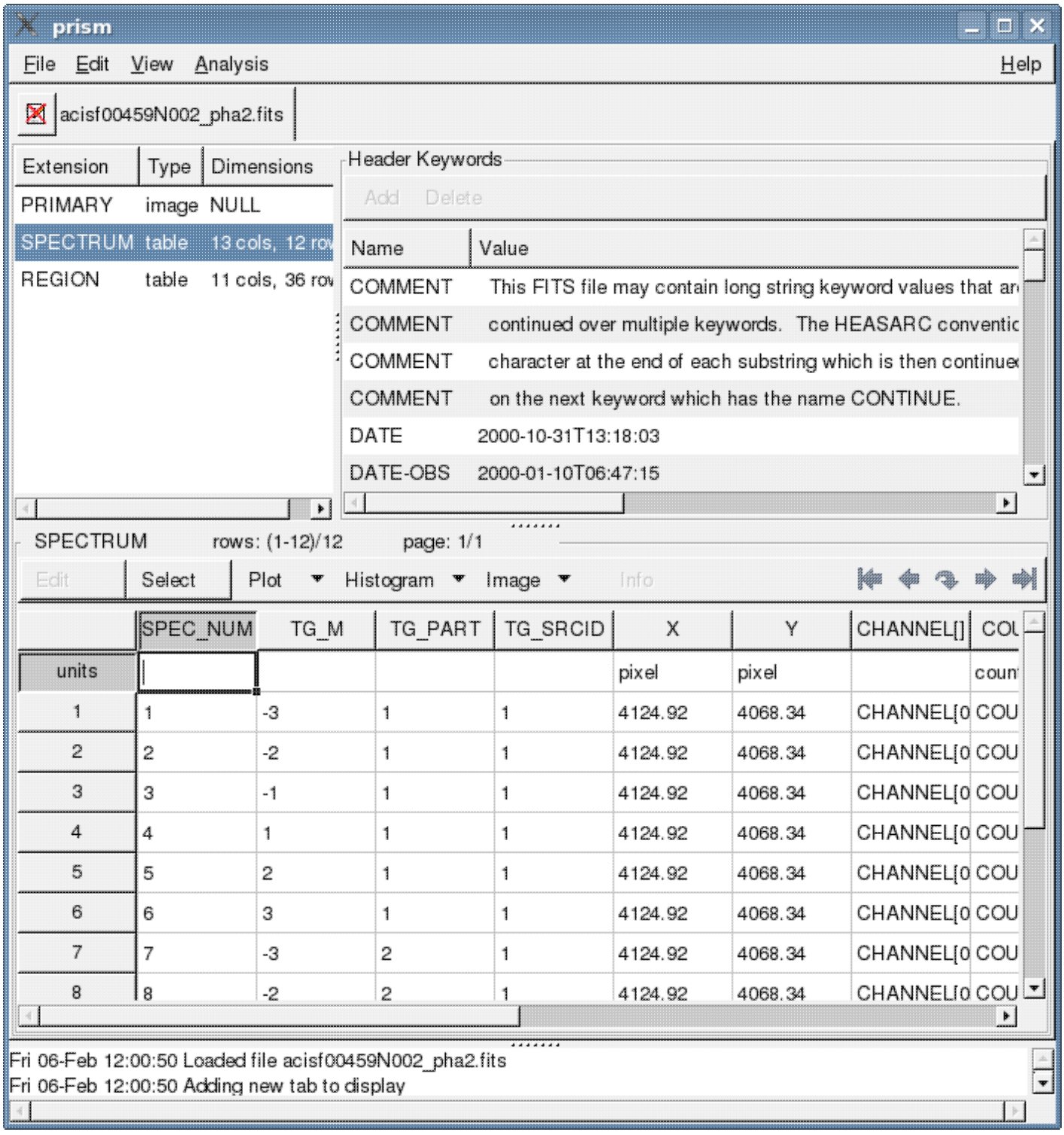


Figure 1: ACIS-S/HETG dataset in prism

Run fullgarf

It is necessary to run `fullgarf` for each order that will be modeled, up to twelve times (once for each row in the PHA file).

For row 1, HEG, order = -3:

```
unix% punlearn fullgarf
unix% pset fullgarf phafile=acisf00459N002_pha2.fits
unix% pset fullgarf pharow=1
unix% pset fullgarf evtfile=acisf00459N002_evt2.fits
unix% pset fullgarf asol=@pcad_asol1.lis
unix% pset fullgarf engrid="grid(rmf.fits[cols ENERG_LO,ENERG_HI])"
unix% pset fullgarf dtffile=")evtfile"
unix% pset fullgarf badpix=acis_459_new_bpix1.fits
unix% pset fullgarf maskfile=acisf00459_000N002_msk1.fits
unix% pset fullgarf pbkfile=acisf063875928N002_pbk0.fits
unix% pset fullgarf dafile=CALDB
unix% pset fullgarf rootname=acisf00459
```

In many cases there will be more than one asol1.fits file for an observation. **All** the files must be input to the `asol` parameter, either as a list or as a stack (see ahelp stack for more information). For example, here we used:

```
unix% more pcad_asol1.lis
pcadf063874624N002_asol1.fits
pcadf063875522N002_asol1.fits
pcadf063902942N002_asol1.fits
```

Now run the tool:

```
unix% fullgarf
Will use /home/username/cxcds_param4/fullgarf.par for the parameter file.

/home/username/cxcds_param4/fullgarf.par contains 15 parameters . . .

Input PHA file (Type I or II) (acisf00459N002_pha2.fits):
Row in Type II PHA file (ignored if Type I) (0:99) (1):
Event file  (acisf00459N002_evt2.fits):
Aspect offsets file (@pcad_asol1.lis):
Energy grid spec (grid(rmf.fits[cols ENERG_LO,ENERG_HI])):
Dead time correction factor; ACIS->evt file; HRC -> dtf file)evtfile -> acisf00459N002_evt2.fits):
Bad pixel file; (filename|NONE|CALDB) (acis_459_new_bpix1.fits):
Output rootname (acisf00459):
NONE, or the name of the parameter block file (acisf063875928N002_pbk0.fits):
NONE, or name of ACIS window mask file (acisf00459_000N002_msk1.fits):
Getting the pha file type . . .

Grating arm is HEG, order=-3
Source location is X=4124.919921875, Y=4068.340087890625

Detector is ACIS
Will run asphist for ccd_id= 4 5 6 7


asphist infile=@pcad_asol1.lis outfile=acisf00459_ah4.fits evtfile=acisf00459N002_evt2.fits[ccd_id=4
dtffile=acisf00459N002_evt2.fits

asphist infile=@pcad_asol1.lis outfile=acisf00459_ah5.fits evtfile=acisf00459N002_evt2.fits[ccd_id=5
dtffile=acisf00459N002_evt2.fits
```

```
asphist infile=@pcad_asol1.lis outfile=acisf00459_ah6.fits evtfile=acisf00459N002_evt2.fits[ccd_id=6]
dtffile=acisf00459N002_evt2.fits

asphist infile=@pcad_asol1.lis outfile=acisf00459_ah7.fits evtfile=acisf00459N002_evt2.fits[ccd_id=7]
dtffile=acisf00459N002_evt2.fits

Finished creating aspect histograms for ccd_id= 4 5 6 7
Will run mkgarf for the same ccd_id list

mkgarf detsubsys=ACIS-S0 order=-3 grating_arm=HEG outfile=acisf00459_S0_HEG_-3.fits asphistfile=acisf00459
[ASPHIST] engrid=grid(rmf.fits[cols ENERG_LO,ENERG_HI]) osipfile=CALDB dafile=CALDB
pbkfile=acisf063875928N002_pbk0.fits maskfile=acisf00459_000N002_msk1.fits mode=hl verb=0
*** ERROR: aspect histogram acisf00459_ah4.fits[ASPHIST] contains no rows

mkgarf detsubsys=ACIS-S1 order=-3 grating_arm=HEG outfile=acisf00459_S1_HEG_-3.fits asphistfile=acisf00459
[ASPHIST] engrid=grid(rmf.fits[cols ENERG_LO,ENERG_HI]) osipfile=CALDB dafile=CALDB
pbkfile=acisf063875928N002_pbk0.fits maskfile=acisf00459_000N002_msk1.fits mode=hl verb=0

mkgarf detsubsys=ACIS-S2 order=-3 grating_arm=HEG outfile=acisf00459_S2_HEG_-3.fits asphistfile=acisf00459
[ASPHIST] engrid=grid(rmf.fits[cols ENERG_LO,ENERG_HI]) osipfile=CALDB dafile=CALDB
pbkfile=acisf063875928N002_pbk0.fits maskfile=acisf00459_000N002_msk1.fits mode=hl verb=0

mkgarf detsubsys=ACIS-S3 order=-3 grating_arm=HEG outfile=acisf00459_S3_HEG_-3.fits asphistfile=acisf00459
[ASPHIST] engrid=grid(rmf.fits[cols ENERG_LO,ENERG_HI]) osipfile=CALDB dafile=CALDB
pbkfile=acisf063875928N002_pbk0.fits maskfile=acisf00459_000N002_msk1.fits mode=hl verb=0
  *** WARNING: The ARF was computed to be zero at all the specified energies.
             This is probably due to an incorrect source position


Finished creating grating arfs for ccd_id= 4 5 6 7

dmarfadd @thisfile.istemporary acisf00459HEG_-3_garf.fits

fullgarf finished.
```

The aspect-related error from `mkgarf` may be ignored; it is explained in this FAQ. Read the "ARF was computed to be zero at all energies" section for an explanation of the final warning.

The content of the parameter file may be checked using plist fullgarf.

The script is run in a similar way for the remaining rows.

Fitting

At this point, you should have the spectra, gARFs, and gRMFs necessary for fitting the data. The Fitting Grating Data thread (S-Lang or Python) shows how to load the data and responses, define a model, and fit the spectra.

In order to use Gaussian statistics to fit a model to a dataset, it is often necessary to "group" the data - i.e. combine channels until you have enough counts. Before fitting the data in *Sherpa*, read the Grouping a Grating Spectrum thread for more information.

The "ARF was computed to be zero at all energies" Warning

When running `mkgarf`, a warning of this form may be printed:

```
*** WARNING: The ARF was computed to be zero at all the specified energies.
             This is probably due to an incorrect source position
```

There are a few possible situations that are causing this:

1. An incorrect source position was input to the `sourcepixelx` and `sourcepixely` parameters. Confirm that the source position is correct, and re-run the tool if it's not.
2. You are attempting to create a gARF for an order that does not fall on that plate, e.g. there is only a bit of negative orders on ACIS-S3.
3. There are some cases where this happens because the mask truncates the order on a chip.

If none of these items resolves the problem, contact the Helpdesk for assistance.

```
Parameters for /home/username/cxcds_param/fullgarf.par


       phafile = acisf00459N002_pha2.fits Input PHA file (Type I or II)
        pharow = 1                Row in Type II PHA file (ignored if Type I)
       evtfile = acisf00459N002_evt2.fits Event file
          asol = @pcad_asol1.lis Aspect offsets file
        engrid = grid(rmf.fits[cols ENERG_LO,ENERG_HI]) Energy grid spec
       dtffile = )evtfile -> acisf00459N002_evt2.fits Dead time correction factor; ACIS->evt file; H
        badpix = acis_459_new_bpix1.fits Bad pixel file; (filename|NONE|CALDB)
      rootname = acisf00459       Output rootname
      maskfile = acisf00459_000N002_msk1.fits NONE, or name of ACIS window mask file
      pbkfile  = acisf063875928N002_pbk0.fits  NONE, or the name of the parameter block file
       (dafile = CALDB)            NONE, CALDB, or name of ACIS dead-area calibration file
     (osipfile = CALDB)           NONE or Name of fits file with order sorting info
      (clobber = no)              Clobber existing output files? This is passed to ALL child process
      (verbose = 0)               Control the level of diagnostic output. 0=>least.
         (mode = hl)              Mode flags.  Set to 'ql' to enable querying, 'hl' to suppress.
```

History

23 Dec 2004	updated for CIAO 3.2: canned gRMFs are no longer available in the CALDB, removed "Choosing an RMF" section
06 Dec 2005	updated for CIAO 3.3: the `fullgarf` script has been updated to version 3.3.1 for the new `asphist` tool syntax; corresponding changes to screen output
01 Dec 2006	reviewed for CIAO 3.4: no changes
26 Feb 2007	fullgarf v4.0.1: Four new parameters have been added, all of which are used by the `mkgarf` tool: `pbkfile`, `dafile`, `osipfile`, and `maskfile`; see "ahelp fullgarf" for details on each new parameter. Added ACIS dead area correction section

11 Jan 2008	updated for CIAO 4.0: `fullgarf` v4.1.1 (parameter updates for applying ACIS dead area correction, which is turned on by default); ACIS QE degradation link moved to "Related Links" section; removed outdated calibration information ; added "ARF was computed to be zero at all energies" section
30 May 2008	fullgarf v4.1.3 (Unix head and tail commands replaced by `pget` and `dmkeypar`); thread images are inline
12 Feb 2009	updated for CIAO 4.1: "ARDLIB warning ... Assuming the first "interesting" extension." no longer printed; input data must have a CTI_APP keyword
12 Feb 2009	updated for CIAO 4.1: "ARDLIB warning ... Assuming the first "interesting" extension." no longer printed; input data must have a CTI_APP keyword
19 Feb 2009	added Fitting section
06 May 2009	check the version of the CIAO scripts package instead of the individual script
01 Jun 2009	fullgarf updated in 01 Jun 2009 scripts package: If bad pixel file is not supplied, use the one from the CALDB. Previously, the script would set the bad pixel file to "NONE" in this case.
16 Jun 2009	added About the Chandra Grating Data Archive and Catalog section

---

URL: http://cxc.harvard.edu/ciao/threads/mkgarf_hetgacis/ Last modified: 16 June 2009