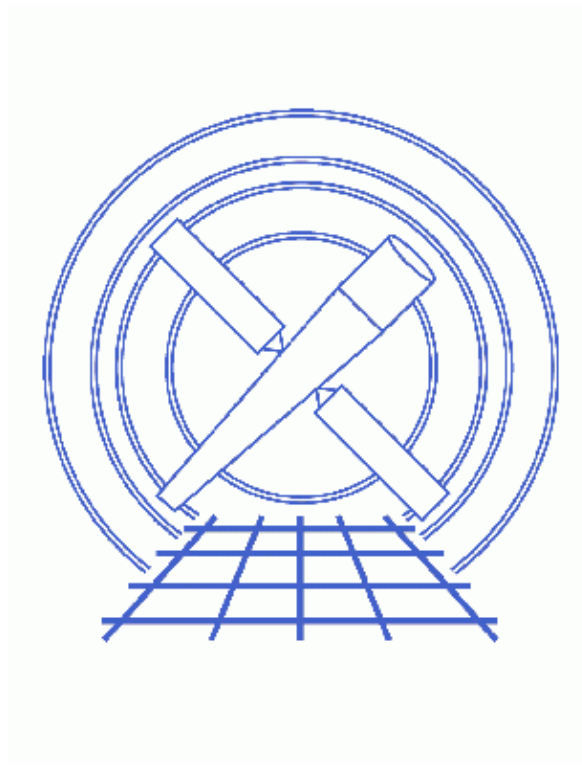


LETG/HRC-S Grating ARFs



CIAO 4.1 Science Threads

Table of Contents

- **About the Chandra Grating Data Archive and Catalog**
- **Background Information**
- **Get Started**
 - ◆ Can't I use fullgarf anymore?
 - ◆ Syntax note: foreach
- **Determine the orders to model**
 - ◆ HRC-S plates and LETG orders
- **Compute the Aspect Histograms (asphist)**
- **Get Source Position (dmlist)**
- **Run mkgarf**
 1. HRC-S1
 2. HRC-S2
 3. HRC-S3
- **Combine the gARFs (dmarfadd)**
- **Fitting**
- **The "ARF was computed to be zero at all energies" Warning**
- **Note on the Effective Area**
- **History**
- **Images**
 - ◆ Figure 1: HRC-S/LETG dataset in prism

LETG/HRC-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:

`mkgarf` creates a grating [ARF](#) for a particular order of an observation. The tool operates on a single detector element (i.e. one HRC plate) and grating order at a time. Therefore, it is necessary to generate all the individual component gARFs and then combine them with `dmarfadd` to create a grating ARF that covers the region of interest for that order.

Purpose:

To create grating ARFs for several orders of an LETG/HRC-S observation.

Read this thread if:

you are working with an LETG/HRC-S observation and intend to model and fit the spectra in *Sherpa*.

Related Links:

- [Analysis Guide for Chandra High Resolution Spectroscopy](#): an in-depth discussion of grating analysis.

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 flux 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.

Background Information

`mkgarf` incorporates the telescope dither explicitly and averages over the response of the detector region which is exposed to a point in the sky. Hence, plate gaps and other detector non-uniformities, as well as the mirror vignetting function, are included. Arbitrary dither, off-axis response, or off-nominal source positions are not accounted for in the pre-computed effective area files.

The tool computes the efficiency for a full aperture in the cross-dispersion direction. The effect of the finite region width is calculated by `mkgrmf`, since the line-spread-function is physically coupled to the extraction width. This aperture efficiency factor is accounted for in analysis when using an ARF and RMF jointly to model the instrument. *Sherpa*, ISIS, and XSPEC will include this factor automatically when using both the ARF and RMF. ISIS also provides a `factor_rsp` function with which this energy-dependent term can be extracted, if it is explicitly needed.

Get Started

Sample ObsID used: 460 (LETG/HRC-S, 3C 273)

File types needed: evt2; pha2; asol1; bpix1; dtf1

It is assumed that you have created gRMFs for your observation by running the [HRC Grating RMFs thread](#).

If you created a new bad pixel file by running the [New Observation-Specific HRC Bad Pixel File thread](#), make sure that you have [set up ardlb](#) to use the same bad pixel file.

Can't I use `fullgarf` anymore?

A previous version of this thread simply used the `fullgarf` script to generate the gARFs. While the script correctly creates +/- 1 gARFS, the functionality does not yet exist to create higher orders. Until `fullgarf` is enhanced to handle those higher orders, it is recommended that users generate the gARFS "manually" by following this thread.

Syntax note: `foreach`

This thread uses "foreach" loops to run the same CIAO tool for multiple chips. The syntax is correct for the [csh or tcsh shell](#). If you are using another shell, e.g. bash, change the loop syntax accordingly.

Determine the orders to model

Because of the low energy resolution in the HRC-S, the PHA2 file contains two rows (negative and positive) (Figure 1) containing all the spectral orders. While it is not possible to separate the overlapping orders, CIAO can make response files (gRMFs and gARFs) for higher orders, allowing you to model and fit them.

The screenshot shows the 'prism' software interface. The main window displays a table with the following data:

	SPEC_NUM	TG_M	TG_PART	TG_SRCID	X	Y	CHANNEL[]	COL
units					pixel	pixel		count
1	1	-1	3	1	32842.6	32638.1	CHANNEL[0 COU	
2	2	1	3	1	32842.6	32638.1	CHANNEL[0 COU	

The status bar at the bottom of the window shows: Fri 06-Feb 12:06:27 Loaded file /data/ciao_demo/threads/thread_output/CIAO4.1/examinepha2/hrcf00460N004_pha2.fi

Figure 1: HRC-S/LETG dataset in prism

We choose to focus on the first three positive and negative orders of the spectra. The Higher-order Responses for HRC-S/LETG Spectra thread provides some advice on determining the number of orders to model.

HRC-S plates and LETG orders

There are three plates in the HRC detector: S1, S2, and S3. When the LETG grating is used, the orders are distributed as follows:

- positive orders: S1 and S2
- negative orders: S2 and S3

Further details on this are available from [Table 9.3](#) in the [The Proposer's Observatory Guide](#).

If you are only interested in the positive orders, for instance, it is unnecessary to create gARFs for the S3 plate; the files generated will contain no data. The ["ARF was computed to be zero at all energies" Warning section](#) contains some related information.

Compute the Aspect Histograms (asphist)

The aspect solution files are used to create a binned histogram detailing the aspect history of the observation. An aspect histogram must be created for each plate of the detector you are going to use.

To create a histogram for each of the three plates - S1 (`chip_id=1`), S2 (`chip_id=2`), and S3 (`chip_id=3`) - the tool is run three times:

```
unix% punlearn asphist
unix% pset asphist infile=pcadf063792009N003_asol1.fits
unix% pset asphist dtffile=hrcf00460_000N004_dtf1.fits

unix% foreach f ( 1 2 3 )
foreach? asphist evtfile="hrcf00460N004_evt2.fits[chip_id=$f]" \
          outfile=460_asphist_s{$f}.fits mode=h
foreach? end

# asphist (CIAO 4.1): WARNING: skipping 41 livetime correction records
(from time: 63792195.736228 to time: 63792277.736231)

# asphist (CIAO 4.1): WARNING: skipping 41 livetime correction records
(from time: 63792195.736228 to time: 63792277.736231)

# asphist (CIAO 4.1): WARNING: skipping 41 livetime correction records
(from time: 63792195.736228 to time: 63792277.736231)

unix% ls -l *asphist*
460_asphist_s1.fits
460_asphist_s2.fits
460_asphist_s3.fits
```

In many cases there will be more than one `asol1.fits` file for an observation. **All** the files must be input to the `infile` parameter, either as a list or as a stack (see [ahelp stack](#) for more information).

Many "skipping N livetime correction records" may be printed to the screen. These warnings are common and *do not* indicate a problem.

Get Source Position (dmlist)

The source position is required as one of the inputs to `mkgarf`. This information can be easily obtained from the PHA2 file with `dmlist`:

```
unix% dmlist hrcf00460N004_pha2.fits"[cols x,y]" opt=data
-----
Data for Table Block SPECTRUM
-----
ROW      X                      Y
-----
1         32842.6250          32638.082031250
2         32842.6250          32638.082031250
```

The source is located at (32842.6250, 32638.082031250).

Run mkgarf

Now we have all the information needed to run `mkgarf`. The tool needs to be run **once per plate for each order**; each order falls on two of the plates. Since we are interested in three positive and three negative orders, we need to run the tool **twelve times**:

- S1: +1, +2, +3
- S2: -1, -2, -3, +1, +2, +3
- S3: -1, -2, -3

Again, refer to the [HRC-S plates and LETG orders section](#) if you need assistance determining how many times to run `mkgarf`. Remember that if you run it for the "wrong" plate, the tool will just issue a warning and create an empty file.

1. HRC-S1

First, we make a gARF for each order on plate S1. To simplify the steps, we loop through the three orders, using a variable to change the necessary parameters between runs. Setting the `mode` parameter to "h" tells the tool not to prompt for values.

Be sure that the S1 aspect histogram is used in the `asphistfile` parameter and that `detsubsys=HRC-S1`.

```
unix% punlearn mkgarf
unix% pset mkgarf asphistfile="460_asphist_s1.fits[ASPHIST]"
unix% pset mkgarf obsfile="hrcf00460N004_evt2.fits[EVENTS]"
unix% pset mkgarf detsubsys=HRC-S1
unix% pset mkgarf grating_arm=LEG
unix% pset mkgarf sourcepixelx=32842.6250 sourcepixely=32638.082031250

unix% foreach f ( 1 2 3 )
foreach? mkgarf outfile=garf_460_S1_LEG_{f}.fits order=$f \
          engrid="grid(460_leg_{$f}.grmf[cols ENERG_LO,ENERG_HI])" \
          mode=h
foreach? end
```

2. HRC-S2

The process is repeated for the S2 plate. **Do not** issue "[punlearn](#)" before running the tool again, as only a few parameters need to change. Note that the loop needs to cover the positive and negative orders, as they all fall on S2.

```
unix% pset mkgarf asphistfile="460_asphist_s2.fits[ASPHIST]"
unix% pset mkgarf detsubsys=HRC-S2

unix% foreach f ( -1 -2 -3 1 2 3 )
foreach? mkgarf outfile=garf_460_S2_LEG_{$f}.fits order=$f \
          engrid="grid(460_leg_{$f}.grmf[cols ENERG_LO,ENERG_HI])" \
          mode=h
foreach? end
```

3. HRC-S3

And, finally, gARFS for S3 are created.

```
unix% pset mkgarf asphistfile="460_asphist_s3.fits[ASPHIST]"
unix% pset mkgarf detsubsys=HRC-S3

unix% foreach f ( -1 -2 -3 )
foreach? mkgarf outfile=garf_460_S3_LEG_{$f}.fits order=$f \
          engrid="grid(460_leg_{$f}.grmf[cols ENERG_LO,ENERG_HI])" \
          mode=h
foreach? end
```

The history of each file, stored in the header, contains the complete `mkgarf` command used to create that gARF:

```
unix% dmhistory garf_460_S2_LEG_3.fits mkgarf
mkgarf asphistfile="460_asphist_s2.fits[ASPHIST]"
outfile="garf_460_S2_LEG_3.fits" order="3" sourcepixelx="32842.625"
sourcepixely="32638.08203125" engrid="grid(460_leg_3.grmf[cols
ENERG_LO,ENERG_HI])" obsfile="hrcf00460N004_evt2.fits[EVENTS]"
osipfile="NONE" maskfile="NONE" mirror="hrma" detsubsys="HRC-S2"
grating_arm="LEG" pbkfile="" dafile="CALDB" ardlparfile="ardlib.par"
geompar="geom" verbose="0" clobber="no"
```

Combine the gARFs (dmarfadd)

`dmarfadd` is used to combine the individual gARFs for each order into a single response file. The tool is run six times, once for each order:

```
unix% punlearn dmarfadd
unix% dmarfadd infile="garf_460_S2_LEG_-1.fits,garf_460_S3_LEG_-1.fits" \
          outfile=460_LEG_-1.garf

unix% dmarfadd infile="garf_460_S2_LEG_-2.fits,garf_460_S3_LEG_-2.fits" \
          outfile=460_LEG_-2.garf

unix% dmarfadd infile="garf_460_S2_LEG_-3.fits,garf_460_S3_LEG_-3.fits" \
          outfile=460_LEG_-3.garf

unix% dmarfadd infile="garf_460_S1_LEG_1.fits,garf_460_S2_LEG_1.fits" \
          outfile=460_LEG_1.garf
```

```

unix% dmarfadd infile="garf_460_S1_LEG_2.fits,garf_460_S2_LEG_2.fits" \
        outfile=460_LEG_2.garf

unix% dmarfadd infile="garf_460_S1_LEG_3.fits,garf_460_S2_LEG_3.fits" \
        outfile=460_LEG_3.garf

```

Fitting

At this point, you should have the spectra, gARFs, and gRMFs necessary for fitting the data. The Fitting Multiple Orders of HRC-S/LETG 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. a positive order on HRC-S3. Refer to the [HRC-S plates and LETG orders section](#) for more information.

If neither of these seems to fix the problem, contact the [Helpdesk](#) for assistance.

Note on the Effective Area

The grating ARFs as generated by `mkgarf` are appropriate for events falling within the full 2π steradian solid angle behind the grating. They do not include a correction for the fraction of these events that fall outside the "bowtie" extraction region whence the first-order spectra are selected from the L2 event file (`tgextract`). This fraction is of order 10%, and a correction for this is included in the RMF generated by `mkgrmf`, specifically in the LSFARM calibration file.

Use of ARF and RMF files combined, e.g. within *Sherpa*, XSPEC, or ISIS, will result in the correct effective area being applied. Should they be required for analysis of LETG+HRC-S observations using methods or software that cannot easily combine RMF and ARF files, pre-generated on-axis LETG+HRC-S ARF files that include the bow-tie "extraction efficiency" are available on the [LETG instrument team web page](#).

History

- 20 Dec 2004 updated for CIAO 3.2: canned gARFs are no longer available in the CALDB, users *must* run this thread, removed "Choosing an RMF" section
- 10 Feb 2005 new **Calibration Updates** entry for CALDB 3.0.1
- 08 Apr 2005 rewritten to no longer use the `fullgarf` script; added information on modeling higher orders
- 06 Dec 2005 updated for CIAO 3.3: new `asphist` tool syntax (the GTI filter is associated with the event file instead of the aspect solution file)
- 01 Dec 2006 updated for CIAO 3.4: CIAO version in warning
- 23 Jan 2008 updated for CIAO 4.0: parameter file updates for `mkgarf` (`obsfile` parameter is set to event file instead of aspect histogram file); *Sherpa* links point to 3.4 website; removed outdated calibration updates
- 12 Feb 2009 updated for CIAO 4.1: images are inline; sherpa link points to 4.1 website
- 19 Feb 2009 moved fitting information from Summary into Fitting section
- 16 Jun 2009 added About the Chandra Grating Data Archive and Catalog section

URL: http://cxc.harvard.edu/ciao/threads/mkgarf_letghrcs/

Last modified: 16 June 2009