Introduction to Grating Analysis and Fitting

Antonella Fruscione

2nd Chandra/CIAO Workshop 24 April 2001

A walk through the main grating CIAO 2.1 threads from *http://asc.harvard.edu/ciao/threads/*

Examining Type 2 PHA Files
(examinepha2.thread.html)
⇒ the starting point for grating analysis

Obtain Grating Spectra from HETG/ACIS-S Data
(spectra.hetgacis.html)
⇒ do I have to run this?

Compute HETG/ACIS-S Grating ARFsv
(mkgarf.hetgacis.html)
⇒ always

Fitting Grating Data
(sherpa.grating.html)
⇒ using SHERPA

Extract Coadded and Grouped Nth-Order Source & Background Spectra and ARFs
(add_grating_orders.thread.html)
⇒ i.e. a different approach

CIAO 2.1 Science Threads

Examining PHA2 Files (4 April 2001)

Return to Threads Page.

Contents:

The following examples use data from ObsID 459 (HETG/ACIS-S 3C 273), ObsID 1198 (LETG/ACIS-S, 3C 273), ObsID 460 (LETG/HRC-S, 3C 273) and ObsID 1800 (LETG/HRC-I, PKS2155-304)

A. What is a PHA2 File?

A Type II PHA file (aka PHA2 file) is a standard FITS format in which each row contains array columns. The PHA2 file is a product of standard data processing (aka the pipeline) and is identified by the _pha2.fits extension. In the case that the user has to reprocess the standard data products (e.g. when applying an updated order sorting table), the PHA2 spectrum file is obtained with the `tgextract` tool from the level 2 event file.

The SPECTRUM block of a PHA2 file has 13 columns of data:

```
ColNo Name                  Description

1     SPEC_NUM              Spectrum Number
2     TG_M                  Diffraction order (m)
3     TG_PART               Spectral component (HEG, MEG, LEG, HESF parts)
4     TG_SRCID              Source ID, output by tgdetect
5     X pixel               X sky coord of source
6     Y pixel               Y sky coord of source
7     CHANNEL[8192]         Vector of spectral bin numbers.
8     COUNTS[8192]          Counts array (a spectrum)
9     STAT_ERR[8192]        Statistical uncertainty (error) on counts colu
10    BACKGROUND_UP[8192]   Upper Background count vector.
11    BACKGROUND_DOWN[8192] Lower Background count vector.
12    BIN_LO[8192]          Bin boundary, left edge
13    BIN_HI[8192]          Bin boundary, right edge
```

The two columns that you need to notice in grating data analysis are TG_M and TG_PART. The TG_M column indicates the order of the spectrum (+/- 1, +/- 2, +/- 3) and the TG_PART column indicates the grating (1 = HEG, 2 = MEG, 3 = LEG).

B. Examining PHA2 Files with *Prism*

1. ACIS-S HETG/LETG Observations

   We can use *Prism* to examine the pha2 file for ObsID 459:

   ```
   unix% prism acism00459N000_pha2.fits &
   ```

   which will give you something like this display. In this example, there are twelve rows - all the +/- orders for both HEG and MEG - for the observation. Note that the columns CHANNEL, COUNTS, BIN_LO, etc. are all so-called "vector columns"; each contains a vector of elements which, in this example, is 8192 elements long.

   Each vector column can be expanded as follows:

   ```
   unix% prism acism00459N000_pha2.fits &

   left mouse click on column of interest (to select column)
   Navigate menu -> Expand Column (to expand column)
   ```

   This is the (partial) result of the expansion of column BIN_LO.

   An ACIS-S/LETG observation (ObsID 1198) looks similar, but only contains 6 rows (+/- orders for the LEG).

2. HRC-S/HRC-I LETG Observations

   Examining an HRC-S/LETG observation (ObsID 460) is done in the same way as an ACIS grating observation:

   ```
   unix% prism hrcm00460N000_pha2.fits &
   ```

but there is an important difference in the results. As seen in the *Prism* display, there are only two rows for the LEG observation. HRC-S cannot resolve orders and the COUNTS in the +/- 1 order are in fact the **total counts** of **all orders combined**. Also, the BIN_LO & BIN_HI columns should be considered for reference only; they actually represent the boundary wavelength of the +/- 1 order alone, while photons from all orders are included in the spectra.

The same holds true for HRC-I/LETG observations, as seen here in the example of ObsID 1800.

C. Displaying the Spectrum

1. With *Prism*

   The quickest way to display one of the spectra of a pha2 file is with *Prism*. For example a user who wants to take a quick look at BIN_LO vs. COUNTS for the order +1 (TG_M=1) of the HEG spectrum (TG_PART=1) should do the following (ObsId 459):

```
unix% prism acism00459N000_pha2.fits &

middle mouse button click on row #4  (to select spectrum)
left mouse click on BIN_LO column  (to select X-axis column)
left mouse click on COUNTS column  (to select Y-axis column)
Vizualization menu -> Quicklook plot (to view the plot)
Vizualization menu -> Print plot (to print the plot)
```

   As clearly visible from the output, the plot is good for a general overview only (are there enough counts, are there striking features, etc.). *Prism* does not allow customization of the plot to date; *ChIPS* should be used for that purpose instead.

2. With *ChIPS*

   In order to display a spectrum with *ChIPS*, we need to separate it out from the PHA2 file; keep in mind that each row of the PHA2 file corresponds to one spectrum. For this we use the CIAO tool `dmtype2split`:

```
unix% dmtype2split infile=acism00459N000_pha2.fits outfile=acism00459N000_h
unix%
```

   The newly created file, which is the order=1 HEG spectrum for this observation, can be used as input to *ChIPS*:

```
unix% chips
chips> plot "acism00459N000_heg_p1.pha[cols bin_lo,counts]"
chips> symbol none
chips> step
chips> xlabel "Wavelength (\AA)"
chips> ylabel Counts
chips> title "ACIS+HEG order=+1"
chips>
```

These commands produce this plot of the data.

An alternative way to read and display a PHA2 spectrum is through S-Lang variables. In this case the PHA2 does not need to be split first. As an example, here are the steps needed to produce the plot above using S-Lang syntax:

```
chips> pha2 = readbintab("acisf00459N0002_pha2.fits")
chips> print(pha2)
filename         =  acisf00459N002_pha2.fits
path             =  /data/ciao/threads/acishetg/from_archive/primary/
filter           =  NULL
ncols            =  13
nrows            =  12
spec_num         =  Short_Type[12]
tg_m             =  Short_Type[12]
tg_part          =  Short_Type[12]
tg_srcid         =  Short_Type[12]
x                =  Float_Type[12]
y                =  Float_Type[12]
channel          =  Short_Type[12,8192]
counts           =  Short_Type[12,8192]
stat_err         =  Float_Type[12,8192]
background_up    =  Short_Type[12,8192]
background_down  =  Short_Type[12,8192]
bin_lo           =  Double_Type[12,8192]
bin_hi           =  Double_Type[12,8192]
chips> bin_lo_heg_p1 = pha2.bin_lo[3,*]
chips> counts_heg_p1 = pha2.counts[3,*]
chips> plot x bin_lo_heg_p1 y counts_heg_p1
chips> symbol none
chips> step
chips> xlabel "Wavelength (\AA)"
chips> ylabel Counts
chips> title "ACIS+HEG order=+1"
chips>
```

Note that S-Lang follows C conventions by numbering array indexes from 0, rather than 1. Therefore the 4th row in the PHA2 file (HEG order +1) is the "3rd" row within the pha2 data arrays (hence the syntax bin_lo_heg_p1 = pha2.bin_lo[3,*] and counts_heg_p1 = pha2.counts[3,*] to identifiy the HEG 1st order).

3. With *Sherpa*

Sherpa can also be used to plot a pha2 spectrum. In this case spectra do not need to be split first - *Sherpa* reads all the rows of a pha2 and allows you to specify individual rows for plotting or fitting purposes:

```
unix% sherpa

-------------------------------------------------
Welcome to Sherpa: CXC’s Modeling and Fitting Program
-------------------------------------------------
Version: 2.0 (2 June 2000)

Type HELP for help options.
Type EXIT, QUIT, or BYE to leave the program.
```

```
Notes:
    Temporary files for visualization will be written to the directory:
    /tmp
    To change this so that these files are not deleted when you exit Sherpa
    edit the variable ASCDS_WORK_PATH in your $HOME/.cxcds.*sh setup script

    Solar abundances set to Anders & Grevesse

sherpa> data acisf00459N002_pha2.fits
The inferred file type is PHA.  If this is not what you want, please
specify the type explicitly in the data command.
The inferred file type is PHA Type II.  If this is not what you want, pleas
specify the type explicitly in the data command.
Failed to find SYS_ERR column
BACKGROUND_UP data are being read from this file.
BACKGROUND_DOWN data are being read from this file.

sherpa> set plot noerrorbars
sherpa> lp 2 data 3 data 4 #(HEG -1 +1)#
sherpa>
```

This plot of the HEG order=-1 (row 3, upper drawing area) and order=+1 (row 4, lower drawing area) is created. Note that no instrument (RMF/ARF) information was given to *Sherpa* and therefore the plot is, by default, COUNTS/SEC vs. CHANNEL (BIN).
To change it to COUNTS vs. CHANNEL the following command can be used:

```
sherpa> ploty 3 counts
sherpa> ploty 4 counts
sherpa> lp 2 data 3 data 4
sherpa>
```

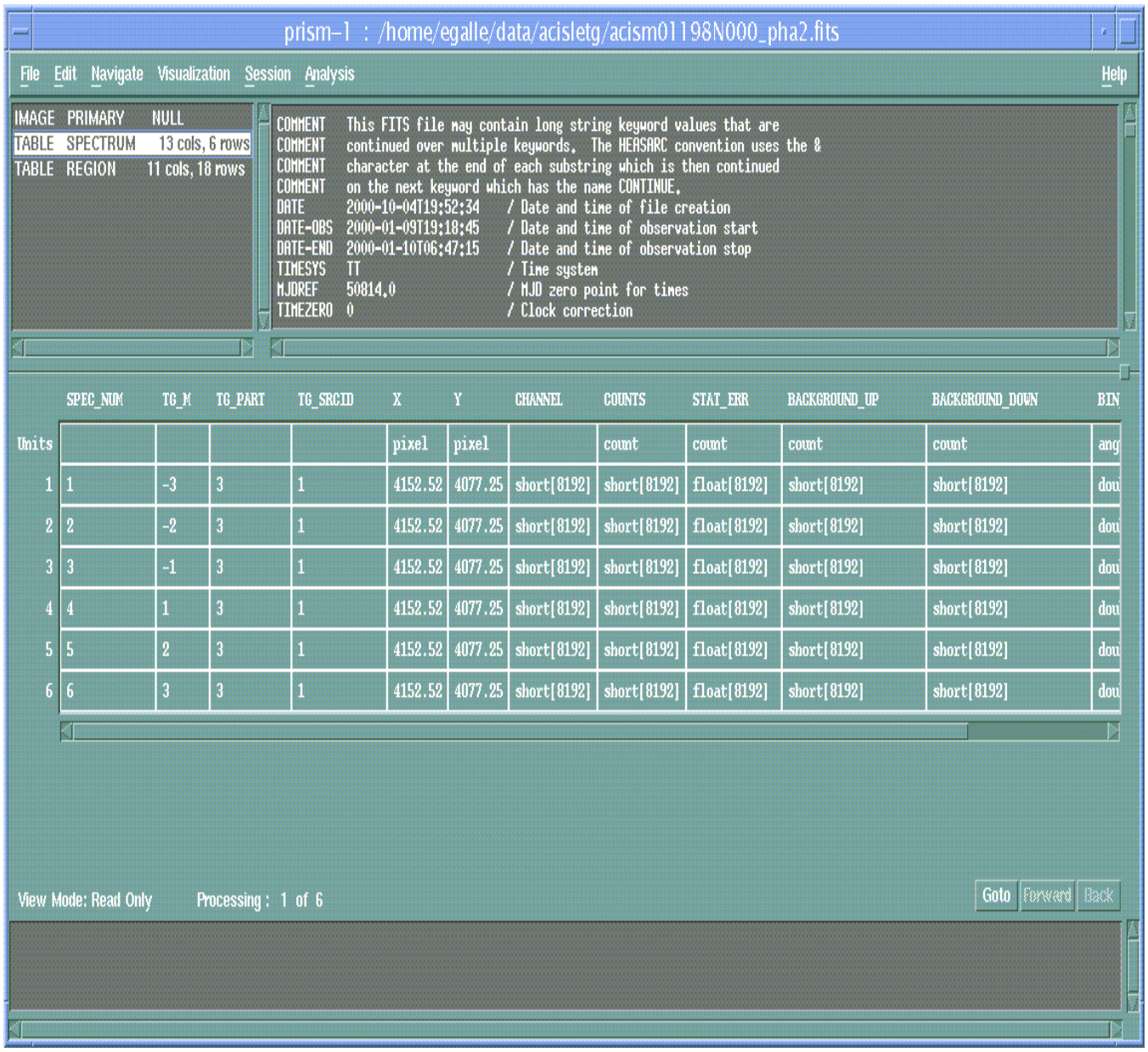
Which alters the plots to look like this. *ChIPS* commands (within *Sherpa*) can be used at this point to customize the plot.

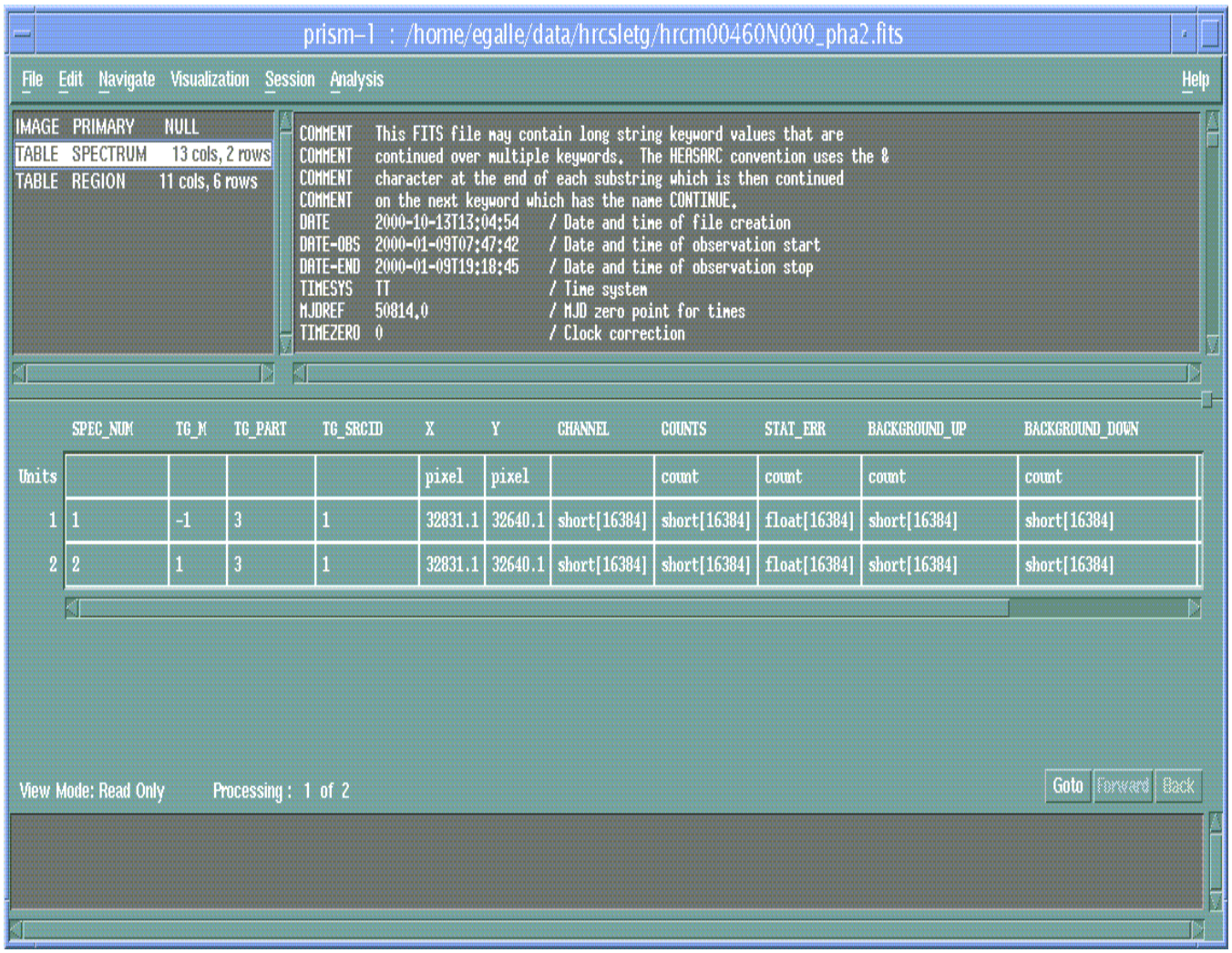
Updates
05 March 2001 -- original version; identical to CIAO 2.0 version.
22 March 2001 -- modified date format
04 April 2001 -- added links back to Threads page

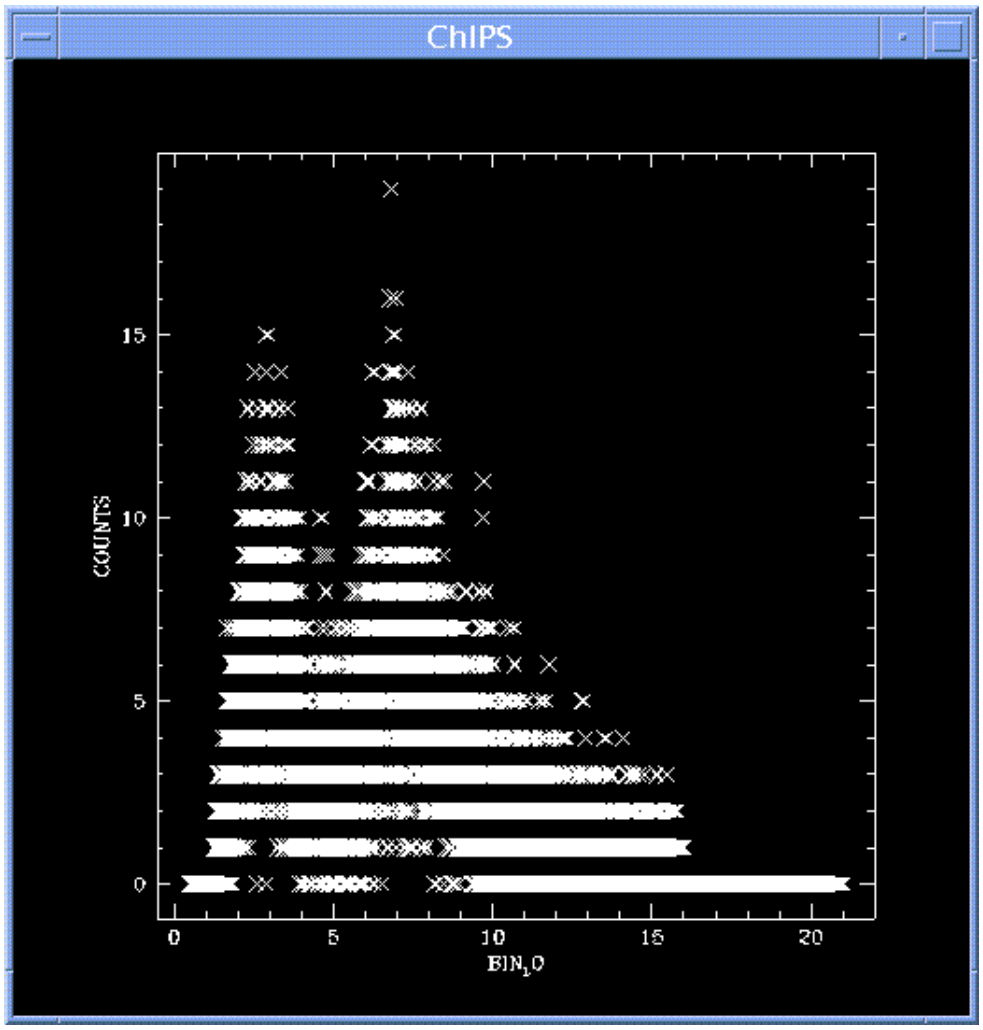
Return to Threads Page.

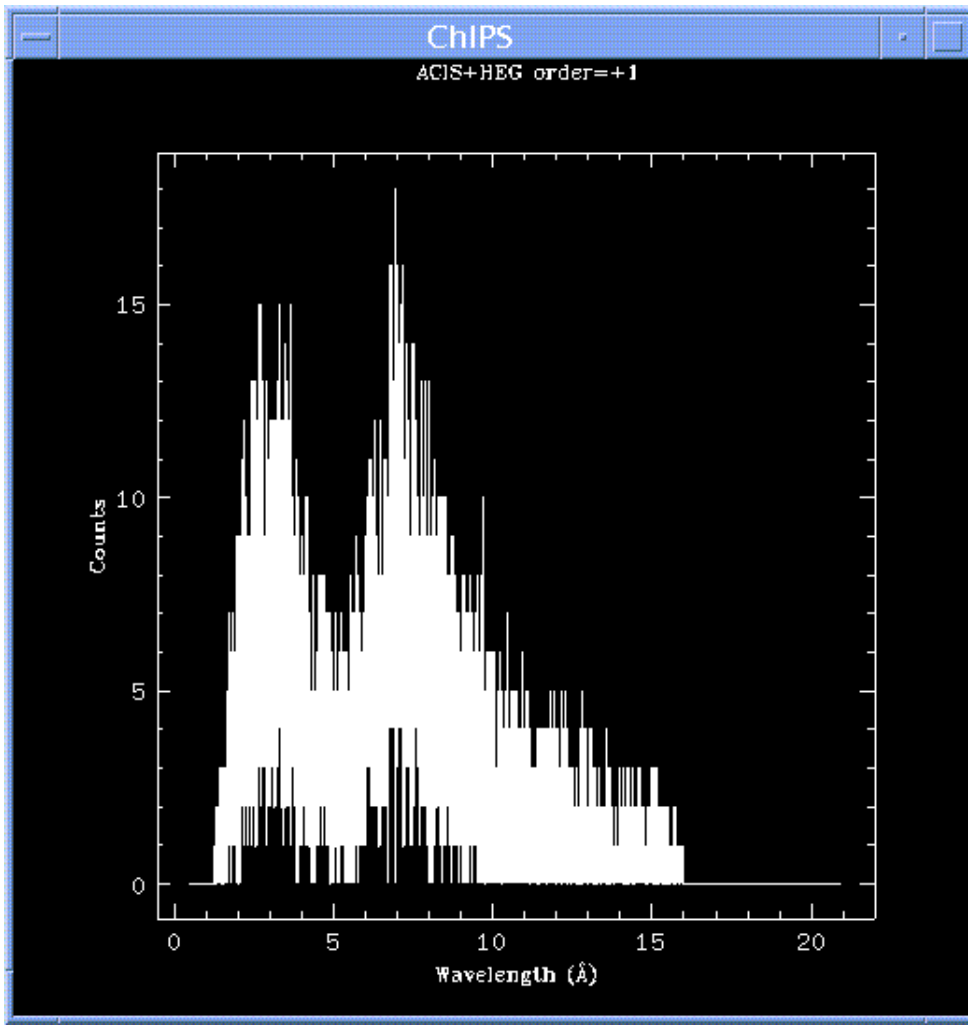
Last modified: 4 April 2001

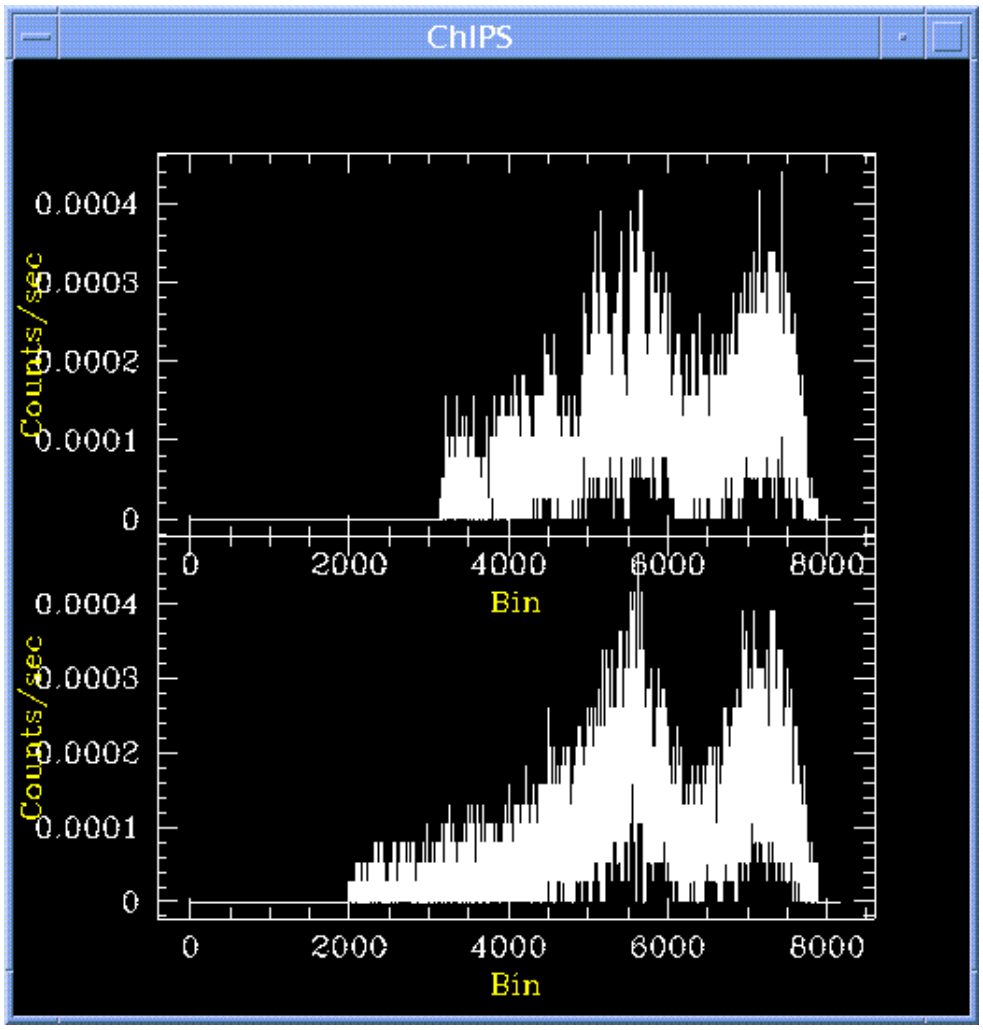

prism-1 : acism00459N000_pha2.fits
File Edit Navigate Visualization Session Analysis Help
IMAGE PRIMARY NULL
TABLE SPECTRUM 13 cols, 12 rows
TABLE REGION 11 cols, 36 rows
COMMENT This FITS file may contain long string keyword values that are
COMMENT continued over multiple keywords. The HEASARC convention uses the &
COMMENT character at the end of each substring which is then continued
COMMENT on the next keyword which has the name CONTINUE.
DATE 2000-10-04T21:59:37 / Date and time of file creation
DATE-OBS 2000-01-10T06:47:15 / Date and time of observation start
DATE-END 2000-01-10T18:23:09 / Date and time of observation stop
TIMESYS TT / Time system
MJDREF 50814.0 / MJD zero point for times
TIMEZERO 0 / Clock correction
SPEC_NUM TG_M TG_PART TG_SRCID X Y CHANNEL COUNTS STAT_ERR BACKGROUND_UP BACKGROUND_DOWN
Units pixel pixel count count count count
1 1 -3 1 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
2 2 -2 1 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
3 3 -1 1 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
4 4 1 1 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
5 5 2 1 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
6 6 3 1 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
7 7 -3 2 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
8 8 -2 2 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
9 9 -1 2 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
10 10 1 2 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
11 11 2 2 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
12 12 3 2 1 4115.14 4061.02 short[8192] short[8192] float[8192] short[8192] short[8192]
View Mode: Read Only
Processing : 11 of 12
Goto Forward Back


prism-1 : /home/egalle/data/acisletg/acism01198N000_pha2.fits
File Edit Navigate Visualization Session Analysis Help
IMAGE PRIMARY NULL
TABLE SPECTRUM 13 cols, 6 rows
TABLE REGION 11 cols, 18 rows
COMMENT This FITS file may contain long string keyword values that are
COMMENT continued over multiple keywords. The HEASARC convention uses the &
COMMENT character at the end of each substring which is then continued
COMMENT on the next keyword which has the name CONTINUE.
DATE 2000-10-04T19:52:34 / Date and time of file creation
DATE-OBS 2000-01-09T19:18:45 / Date and time of observation start
DATE-END 2000-01-10T06:47:15 / Date and time of observation stop
TIMESYS TT / Time system
MJDREF 50814.0 / MJD zero point for times
TIMEZERO 0 / Clock correction
SPEC_NUM TG_M TG_PART TG_SRCID X Y CHANNEL COUNTS STAT_ERR BACKGROUND_UP BACKGROUND_DOWN BIN
Units pixel pixel count count count count count ang
1 1 -3 3 1 4152.52 4077.25 short[8192] short[8192] float[8192] short[8192] short[8192] dou
2 2 -2 3 1 4152.52 4077.25 short[8192] short[8192] float[8192] short[8192] short[8192] dou
3 3 -1 3 1 4152.52 4077.25 short[8192] short[8192] float[8192] short[8192] short[8192] dou
4 4 1 3 1 4152.52 4077.25 short[8192] short[8192] float[8192] short[8192] short[8192] dou
5 5 2 3 1 4152.52 4077.25 short[8192] short[8192] float[8192] short[8192] short[8192] dou
6 6 3 3 1 4152.52 4077.25 short[8192] short[8192] float[8192] short[8192] short[8192] dou
View Mode: Read Only
Processing : 1 of 6
Goto Forward Back


prism-1 : /home/egalle/data/hrcsletg/hrcm00460N000_pha2.fits
File Edit Navigate Visualization Session Analysis Help
IMAGE PRIMARY NULL
TABLE SPECTRUM 13 cols, 2 rows
TABLE REGION 11 cols, 6 rows
COMMENT This FITS file may contain long string keyword values that are
COMMENT continued over multiple keywords. The HEASARC convention uses the &
COMMENT character at the end of each substring which is then continued
COMMENT on the next keyword which has the name CONTINUE.
DATE 2000-10-13T13:04:54 / Date and time of file creation
DATE-OBS 2000-01-09T07:47:42 / Date and time of observation start
DATE-END 2000-01-09T19:18:45 / Date and time of observation stop
TIMESYS TT / Time system
MJDREF 50814.0 / MJD zero point for times
TIMEZERO 0 / Clock correction
SPEC_NUM TG_M TG_PART TG_SRCID X Y CHANNEL COUNTS STAT_ERR BACKGROUND_UP BACKGROUND_DOWN
Units pixel pixel count count count count
1 1 -1 3 1 32831.1 32640.1 short[16384] short[16384] float[16384] short[16384] short[16384]
2 2 1 3 1 32831.1 32640.1 short[16384] short[16384] float[16384] short[16384] short[16384]
View Mode: Read Only Processing : 1 of 2
Goto Forward Back


ChIPS
COUNTS
0
5
10
15
0
5
10
15
20
BIN 0


ChIPS
ACIS+HEG order=+1
Counts
0
5
10
15
0
5
10
15
20
Wavelength (Å)


ChIPS
0.0004
0.0003
0.0002
0.0001
0
Counts/sec
0
2000
4000
6000
8000
Bin
0.0004
0.0003
0.0002
0.0001
0
Counts/sec
0
2000
4000
6000
8000
Bin

CIAO 2.1 Science Threads

Obtain Grating Spectra from HETG/ACIS-S Data (4 April 2001)

Return to Threads Page.

Note: New (corrected) order-sorting tables were added to the processing CALDB in version 1.8.1. The effective date of this addition was 2000-12-07T14:25:00; HETG/ACIS-S files with a header DATE value before this were not processed with the new tables. If your data was processed before this date, run this thread to ensure usage of the latest calibration files (Get Started shows how to check the DATE and CALDBVER).

Also Note: New ACIS gain files have been added to the processing CALDB, which is currently running at version 2.2. Please check the ACIS Gain Map Correction (with option to remove pixel randomization) thread to ensure that your data was processed with the most recent files.

Even if your data was processed after 2000-12-07, you may still wish to follow the last 2 steps (D and E) of this thread to "destreak" the level=2 event file obtained from the pipeline and extract a "cleaner" spectrum; the destreak tool is not part of standard data processing.

If you **do not** run this thread (in particular the Extract a Grating Spectrum step), then you should check the header keywords of the PHA2 file produced by the pipeline as they may contain incorrect values.

For illustration, this thread utilizes data from the ObsID 459 dataset (HETG/ACIS-S, 3C 273), which was downloaded from the Archive. In this example, all level=1 and level=aspect data files were downloaded.

Contents:

2. Apply grade/status filters (**dmcopy**)

D. Run destreak
E. Extract a Grating Spectrum (**tgextract**)

- plist tgdetect
- plist tg_create_mask
- plist tg_resolve_events
- plist destreak
- plist tgextract

The goal of this thread is to generate a new PHA2 spectrum file.

A. Get Started

The following thread is meant to be run with **reprocessed** Chandra data only. If you are working with old data, check the Archive for an updated version of it. Hereafter we will call the level=1 event file:

```
acis_evt1.fits
```

Check the CALDBVER and DATE keywords in the header:

```
unix% dmlist acis_evt1.fits header | grep CALDBVER
0114 CALDBVER              1.6                             String
unix% dmlist acis_evt1.fits header | grep "DATE "
0011 DATE                 2000-10-31T09:28:55              String       Date and t
```

Since this data was processed before 2000-12-07T14:25:00 and with a CALDBVER lower than 1.8.1, we need to complete this thread in its entirety.

If your data was processed with at least CALDB v1.8.1 but you wish to destreak the level=2 file (obtained from the pipeline), skip to the Run destreak step. Otherwise you do not need to run this thread at all and can go directly to the Compute HETG/ACIS-S Grating ARFs thread.

B. Generate A New Level=1.5 Events File

Contents:

1. Get position of zero-order image (**tgdetect**)
2. Get region mask (**tg_create_mask**)
3. Run **tg_resolve_events**

1. Get position of zero-order image (tgdetect)

   To find the zero-order location, the tool tgdetect is run:

CIAO 2.1 Science Threads

Compute HETG/ACIS-S Grating ARFs (4 April 2001)

Return to Threads Page.

For illustration, this threads utilizes data from the ObsID 459 dataset (HETG/ACIS-S, 3C 273), which was downloaded from the Archive. Note that you will need to download all level=1, level=1.5, and level=2 data files for this thread.

The goal of this thread is to generate ARFs by means of a shell script.

Contents:

---

1. Determine the Orders of the Observation

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

   ```
   unix% prism acism00459N000_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). In this example, there are twelve rows (all +/- orders for HEG and MEG) for the HETG observation.

2. Choosing an RMF

   All the necessary RMF files are distributed with the CALDB and are located in the $CALDB/data/chandra/tel/grating/hetg/cpf/rmf. There is an hetgrmf_manifest.txt file in the same directory that explains the RMF naming scheme and which one to use for your analysis (there are

separate RMFs that you must use if you have run the Obtain Grating Spectra from HETG/ACIS-S Data thread).

3. Run mkgarf_acis

Now we have all the information needed to run the script, which is available here. Executing the script without any input files will produce this:

```
unix% mkgarf_acis

USAGE: mkgarf_acis  pha row  rmf  aoff evt2 bpix root

    Compute the ARF for +-'order' for the grating and order found in
    row 'row' of file 'pha' SPECTRUM extension, with  'rmf' for
     the energy grid, and observational files aoff, evt2 (for GTI),
     and bpix.
  Output files will have names 'root_ssss.fits'.

EXAMPLES:

mkgarf_acis  acisf0145N003_pha2.fits  9  acismeg1D1999-07-22rmfN0001.fits acisf(
                 pha2                 row     grating_rmf                    (

 (This is a shell script; you can alter it to suit your own tastes.)
unix%
```

This header shows what files are necessary to run the script. We will have to run the script twelve times, for rows 1 - 12, in this example.

For row 1, HEG, order = -3:

```
unix% mkgarf_acis acism00459N000_pha2.fits 1 caldb/tel/grating/hetg/cpf/rmf/acis

Apply sim component to offsets: asp_apply_sim ...
Make/test for aspect histograms...4  5  6  7  8  9

Get source position from pha2:

Got source position, (x,y) = (      4115.1401367188,      4061.0200195312)

- 1-
1
Got grating type:  'HEG'
Get order from pha2:
Got order  '-3'

Coffee break!  Making chip arfs. This will take a while....
............ acism00459N000_S0_HEG_-3.fits ................
mkgarf detsubsys=ACIS-S0 order=-3 grating_arm=HEG outfile=acism00459N000_S0_HEG_
............ acism00459N000_S1_HEG_-3.fits ................
mkgarf detsubsys=ACIS-S1 order=-3 grating_arm=HEG outfile=acism00459N000_S1_HEG_
............ acism00459N000_S2_HEG_-3.fits ................
mkgarf detsubsys=ACIS-S2 order=-3 grating_arm=HEG outfile=acism00459N000_S2_HEG_
............ acism00459N000_S3_HEG_-3.fits ................
mkgarf detsubsys=ACIS-S3 order=-3 grating_arm=HEG outfile=acism00459N000_S3_HEG_
Done with all gARF pieces.  Add them...
Creating acism00459N000HEG_-3_garf.fits ...
```

```
 All done.  You may want to clean up the gARF pieces,
 if you are satisfied with the results.
unix%
```

The script is run in a similar way for the remaining rows, changing the rmf file to correspond to the correct order (row 2, HEG, order = -2; row 3, HEG, order = -1 ... row 7, MEG, order = -3; row 8, MEG, order = -2, etc).

The thread is now complete. The grating ARFs for this dataset are:

```
unix% ls *garf.fits
acism00459N000HEG_-1_garf.fits  acism00459N000MEG_-1_garf.fits
acism00459N000HEG_-2_garf.fits  acism00459N000MEG_-2_garf.fits
acism00459N000HEG_-3_garf.fits  acism00459N000MEG_-3_garf.fits
acism00459N000HEG_1_garf.fits   acism00459N000MEG_1_garf.fits
acism00459N000HEG_2_garf.fits   acism00459N000MEG_2_garf.fits
acism00459N000HEG_3_garf.fits   acism00459N000MEG_3_garf.fits
```

Updates
05 March 2001 -- original version; identical to CIAO 2.0 version.
22 March 2001 -- modified date format
30 March 2001 -- added the section on Choosing an RMF
04 April 2001 -- added links back to Threads page

Return to Threads Page.

Last modified: 4 April 2001

CIAO 2.1 Science Threads

Fitting Grating Data (6 April 2001)

Return to Threads Page.

Contents:

Getting Started

If you haven't obtained it yet, the data used in this thread is available as sherpa.tar.

Move to the directory that contains the thread data:

```
unix% cd /localpath/threads/sherpa
unix% ls
basic/          fakeit/         grating/        plotting/       spectra/
```

The subdirectory for this thread is named `grating`:

```
unix% cd grating
```

Start a *Sherpa* session:

```
unix% sherpa

-------------------------------------------------------
Welcome to Sherpa: CXC's Modeling and Fitting Program
-------------------------------------------------------
Version: 2.1 (1 Dec 2000)

Type HELP for help options.
Type EXIT, QUIT, or BYE to leave the program.

Notes:
    Temporary files for visualization will be written to the directory:
    /tmp
    To change this so that these files are not deleted when you exit Sherpa,
    edit the variable ASCDS_WORK_PATH in your $HOME/.cxcds.*sh setup script.

    Solar abundances set to Anders & Grevesse

sherpa> pwd
/chandra/threads/sherpa/grating
```

The contents of this directory may be examined, using the Unix command `ls`:

```
sherpa> ls
acism00459N000HEG_-1_garf.fits  session1.sherpascript
acism00459N000HEG_1_garf.fits   sherpa.grating.1.ps
acism00459N000MEG_-1_garf.fits  sherpa.grating.2.ps
acism00459N000MEG_1_garf.fits   sherpa.grating.3.ps
acism00459N000_pha2.fits        thread
session1.ps                     thread.fitresults
```

Fitting grating data typically involves both the PHA Type II datafile and the corresponding instrument Ancillary Response Files (ARF). In this thread, these files are:

`acism00459N000_pha2.fits`, and
`acism00459N000HEG_-1_garf.fits`,
`acism00459N000HEG_1_garf.fits`,
`acism00459N000MEG_-1_garf.fits`, and
`acism00459N000MEG_1_garf.fits`.

To view the contents of the `thread` ASCII file, which contains all of the commands that will be issued during this thread, the Unix command `more` may be used as follows:

```
sherpa> $more thread
### grating Thread:
.
```

Reading FITS PHA Type II Data Into *Sherpa*

In this thread, we fit grating data from the FITS PHA Type II datafile `acism00459N000_pha2.fits`.

This dataset is input into *Sherpa* with the READ command:

```
sherpa> DATA acism00459N000_pha2.fits
The inferred file type is PHA.  If this is not what you want, please
specify the type explicitly in the data command.
The inferred file type is PHA Type II.  If this is not what you want, please
specify the type explicitly in the data command.
Failed to find SYS_ERR column
BACKGROUND_UP data are being read from this file.
BACKGROUND_DOWN data are being read from this file.
```

Note that the READ DATA command may be shortened to just DATA.

The command SHOW returns the current *Sherpa* status; from running SHOW we see that twelve datasets were read from the FITS PHA Type II datafile. This thread will fit the four datasets numbered 3, 4, 9, and 10.

```
sherpa> SHOW
```

Reading Multiple Instrument Responses Into *Sherpa*

Before establishing any models, we first wish to turn off model parameter prompting:

```
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
```

The *Sherpa* model name for an instrument response model, defined by the standard instrument response files, is RSP. Next, the RSP model component is established for use for the four datasets of interest:

```
sherpa> RSP[hegm1]
sherpa> RSP[hegp1]
sherpa> RSP[megm1]
sherpa> RSP[megp1]
```

where hegm1 will be the name of one dataset's instrument response model, hegp1 will be the name of another dataset's instrument response model, etc.

Next, we set the arf parameter values of these models to the proper ARF filenames for each of the four datasets:

```
sherpa> hegm1.arf = acism00459N000HEG_-1_garf.fits
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.
sherpa> hegp1.arf = acism00459N000HEG_1_garf.fits
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.
sherpa> megm1.arf = acism00459N000MEG_-1_garf.fits
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.
sherpa> megp1.arf = acism00459N000MEG_1_garf.fits
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.
```

In order to convolve the input datasets with the response model components that have been established, they must be defined as the instrument models. In this thread, we wish to fit datasets 3, 4, 9, and 10, using response model components hegm1, hegp1, megm1, and mebp1, respectively:

```
sherpa> INSTRUMENT 3 = hegm1
sherpa> INSTRUMENT 4 = hegp1
sherpa> INSTRUMENT 9 = megm1
sherpa> INSTRUMENT 10= megp1
```

The current definition of *Sherpa*'s instrument model for say, dataset number 3, may be examined using `SHOW INSTRUMENT 3`:

```
sherpa> SHOW INSTRUMENT 3
rsp[hegm1]
  RMF file name:    ARF file name: "acism00459N000HEG_-1_garf.fits " (N_E=8192)
EEARF file name:
```

This output shows that `acism00459N000HEG_-1_garf.fits` currently defines the instrument model for dataset number 3. The instrument model definitions for the other datasets may be similarly examined using

```
SHOW INSTRUMENT #
```

where # indicates the dataset number.

The input datasets may now be plotted:

```
sherpa> SET PLOT NOERRORBARS
sherpa> LPLOT 4 DATA 3 DATA 4 DATA 9 DATA 10
```

where `LPLOT 4` specifies four drawing areas, and each `DATA` portion of the `LPLOT` command specifies the dataset number. The resulting plot displays datasets 3, 4, 9, and 10 starting with the uppermost drawing area, respectively.

 Figure 1 shows the resulting plot.

Filtering Grating Data

Before filtering the data, the `ANALYSIS` command is used to check the units with which the data are to be analyzed in this *Sherpa* session:

```
sherpa> ANALYSIS
Analysis Space for Dataset 1: Wavelength
Analysis Space for Dataset 2: Wavelength
Analysis Space for Dataset 3: Wavelength
Analysis Space for Dataset 4: Wavelength
Analysis Space for Dataset 5: Wavelength
Analysis Space for Dataset 6: Wavelength
Analysis Space for Dataset 7: Wavelength
Analysis Space for Dataset 8: Wavelength
Analysis Space for Dataset 9: Wavelength
Analysis Space for Dataset 10: Wavelength
Analysis Space for Dataset 11: Wavelength
```

```
Analysis Space for Dataset 12: Wavelength
```

As expected from this thread's input FITS PHA Type II dataset, the units are wavelength. Therefore, we will specify filter constraints within *Sherpa* in terms of wavelength:

```
sherpa> IGNORE ALLSETS ALL
sherpa> NOTICE 3 WAVE 1.2 : 15
sherpa> NOTICE 4 WAVE 1.2 : 15
sherpa> NOTICE 9 WAVE 2.5 : 31
sherpa> NOTICE 10 WAVE 2.5 : 31
```

Notice that we first filter out all data from all datasets using the command `IGNORE ALLSETS ALL`. Then, we filter by inclusion the appropriate wavelength region for each dataset.

Each filtered dataset may then be plotted:

```
sherpa> SET PLOT NOERRORBARS
sherpa> LPLOT 4 DATA 3 DATA 4 DATA 9 DATA 10
```

Notice that the plot now includes only the data in the specified wavelength regions.

 Figure 2 shows the resulting plot.

Establishing Model Components

In this thread we wish to fit these grating data using source and background model expressions that involve multiple model components.

The first of these model components is the broken power-law model, `BPL`. The `BPL` model component is established for use and is named `bplmh`:

```
sherpa> BPL[bplmh]
```

Note that since a dataset has been previously input, *Sherpa* estimates the initial parameter values (and the minimum and maximum for their ranges) for this model based on the data.

The second of the model components is the one-dimensional power-law model, `POWLAW1D`. The `POWLAW1D` model component is established for use and is named `powbkgmh`:

```
sherpa> POWLAW1D[powbkgmh]
```

The third and final model component is the ISM attenuation model, `ATTEN`. The `ATTEN` model component is established for use and is named `abs`:

```
sherpa> ATTEN[abs]
```

Defining Multi-Component Source Model Expressions

In this thread we wish to fit these grating data using an expression that is the product of two of the model components that we have just established. The source model expression to be used for fitting datasets 3, 4, 9, and 10 is defined as follows:

```
sherpa> SOURCE 3,4,9,10 = abs*bplmh
```

The current definition of *Sherpa*'s source model for say, dataset number 3, may be examined using `SHOW SOURCE 3`:

```
sherpa> SHOW SOURCE 3
(abs * bplmh)
Atten[abs]  (integrate: off)
    Param   Type      Value        Min        Max                Units
    -----   ----      -----        ---        ---                -----
 1   hcol thawed      1e+20      1e+17      1e+24
 2heiRatio thawed     0.1000          0          1
 3heiiRatio thawed     0.0100          0          1
bpl1d[bplmh]  (integrate: on)
    Param   Type      Value        Min        Max                Units
    -----   ----      -----        ---        ---                -----
 1 gamma1 thawed          0        -10         10
 2 gamma2 thawed          0        -10         10
 3     eb thawed        100          0       1000
 4    ref frozen          1-3.4028e+38 3.4028e+38
 5   ampl thawed          1      1e-20 3.4028e+38
```

This output shows that `abs * bplmh` is currently defined as this dataset's source model expression. By default, integration is turned off for the `abs` model component and is turned on for the `bplmh` model component. However, since we are using a multiplicative source model expression, integration *will* be performed during fitting.

The source model definitions for the other datasets may be similarly examined using
`SHOW SOURCE #`
where # indicates the dataset number; these outputs should all be identical since all four datasets are to be fit with the same source model expression (*i.e.* `abs * bplmh`).

Defining Multi-Component Background Model Expressions

We wish to simulatenously fit the background of these grating data, using another expression that is also the product of two of the model components that we have just established. The background model expression to be used for fitting datasets 3, 4, 9, and 10 is defined as follows:

```
sherpa> BACKGROUND 3,4,9,10 = abs*powbkgmh
```

The current definition of *Sherpa*'s background model for say, dataset number 3, may be examined using `SHOW BACKGROUND 3`:

```
sherpa> SHOW BACKGROUND 3
(abs * powbkgmh)
Atten[abs]  (integrate: off)
    Param   Type      Value        Min        Max                Units
```

```
    -----   ----      -----        ---        ---                 -----
 1   hcol thawed      1e+20      1e+17      1e+24
 2heiRatio thawed     0.1000          0          1
 3heiiRatio thawed     0.0100          0          1
powlaw1d[powbkgmh]  (integrate: on)
    Param   Type      Value        Min        Max                 Units
    -----   ----      -----        ---        ---                 -----
 1  gamma thawed          0        -10         10
 2    ref frozen          1-3.4028e+38 3.4028e+38
 3   ampl thawed          1      1e-20 3.4028e+38
```

The background model definitions for the other datasets may be similarly examined using
`SHOW BACKGROUND #`
where # indicates the dataset number; these outputs should all be identical since all four datasets are to be fit with the same background model expression (*i.e.* `abs * powbkgmh`).

Examining Method & Statistic Settings

The `SHOW` command may be used to examine the current method and statistics settings:

```
sherpa> SHOW METHOD
Optimization Method: Powell

      Name       Value         Min         Max                     Description
      ----       -----         ---         ---                     -----------
 1 verbose           1           0           5                 Verbosity level
 2   iters         300           1       10000    Maximum number of iterations
 3     eps       1e-06       1e-09      0.0010             Fractional accuracy
 4     tol       1e-06       1e-08      0.1000             Tolerance in lnmnop
 5    huge       1e+10        1000       1e+12             Fractional accuracy

sherpa> SHOW STATISTIC
Statistic: Chi-Squared Gehrels
```

Further information is available by typing `AHELP POWELL`, and `AHELP CHIGEHRELS`. For this thread, we do not wish to modify these method and statistic settings.

Modifying Initial Model Parameter Values

From the previous `SHOW SOURCE 3` output we see the initial source model parameter values. However, we wish to change a number of these parameter values before fitting:

```
sherpa> bplmh.gamma1 = 0.4
sherpa> bplmh.gamma2 = -0.03
sherpa> bplmh.eb = 13.19
sherpa> bplmh.ref = 8
sherpa> bplmh.ampl = 0.00211901
sherpa> SHOW bplmh
bpl1d[bplmh]  (integrate: on)
    Param   Type      Value        Min        Max                 Units
    -----   ----      -----        ---        ---                 -----
```

```
 1 gamma1 thawed     0.4000        -10         10
 2 gamma2 thawed    -0.0300        -10         10
 3     eb thawed    13.1900          0       1000
 4    ref frozen          8-3.4028e+38 3.4028e+38
 5   ampl thawed     0.0021      1e-20 3.4028e+38

sherpa> powbkgmh.gamma = 0.2
sherpa> powbkgmh.ref = 8
sherpa> powbkgmh.ampl = 2.11438e-05
sherpa> SHOW powbkgmh
powlaw1d[powbkgmh]  (integrate: on)
    Param   Type      Value        Min        Max                Units
    -----   ----      -----        ---        ---                -----
 1  gamma thawed     0.2000        -10         10
 2    ref frozen          8-3.4028e+38 3.4028e+38
 3   ampl thawed     0.0000      1e-20 3.4028e+38

sherpa> abs.hcol = 1.81e20
sherpa> abs.heiRatio = 0.1
sherpa> abs.heiiRatio = 0.01
sherpa> SHOW abs
Atten[abs]  (integrate: off)
    Param   Type      Value        Min        Max                Units
    -----   ----      -----        ---        ---                -----
 1   hcol thawed 1.8100e+20      1e+17      1e+24
 2heiRatio thawed     0.1000          0          1
 3heiiRatio thawed     0.0100          0          1
```

In addition, we wish to freeze all of the `abs` parameters:

```
sherpa> FREEZE abs
sherpa> SHOW abs
Atten[abs]  (integrate: off)
    Param   Type      Value        Min        Max                Units
    -----   ----      -----        ---        ---                -----
 1   hcol frozen 1.8100e+20      1e+17      1e+24
 2heiRatio frozen     0.1000          0          1
 3heiiRatio frozen     0.0100          0          1
```

Fitting

The datasets are then fit:

```
sherpa> FIT
WARNING: Dataset 1 will not be included in the fit.
WARNING: Dataset 2 will not be included in the fit.
WARNING: Dataset 5 will not be included in the fit.
WARNING: Dataset 6 will not be included in the fit.
WARNING: Dataset 7 will not be included in the fit.
WARNING: Dataset 8 will not be included in the fit.
WARNING: Dataset 11 will not be included in the fit.
WARNING: Dataset 12 will not be included in the fit.
 powll: v1.2
 powll:   initial function value =     8.93190E+03
 powll:     converged to minimum =     8.59123E+03 at iteration =      5
 powll:   final function value    =     8.59123E+03
         bplmh.gamma1  0.420342
```

```
      bplmh.gamma2  -0.0296031
      bplmh.eb  18.4653
      bplmh.ampl  0.00197857
   powbkgmh.gamma  0.221657
   powbkgmh.ampl  1.7893e-05
```

Note that the fit of these data may be very time consuming; it takes about 15 minutes on a Sun Ultra60. If you do not want to wait, abort the fitting session using `cntrl-c` and exit *Sherpa*. You may then start another *Sherpa* session and load in previously saved fit results:

cntrl-c
sherpa> EXIT
unix% sherpa
sherpa> USE thread.fitresults

To plot the fits:

```
sherpa> SET PLOT NOERRORBARS
sherpa> LPLOT 4 FIT 3 FIT 4 FIT 9 FIT 10
```

This plot may be improved using the following *ChIPS* commands:

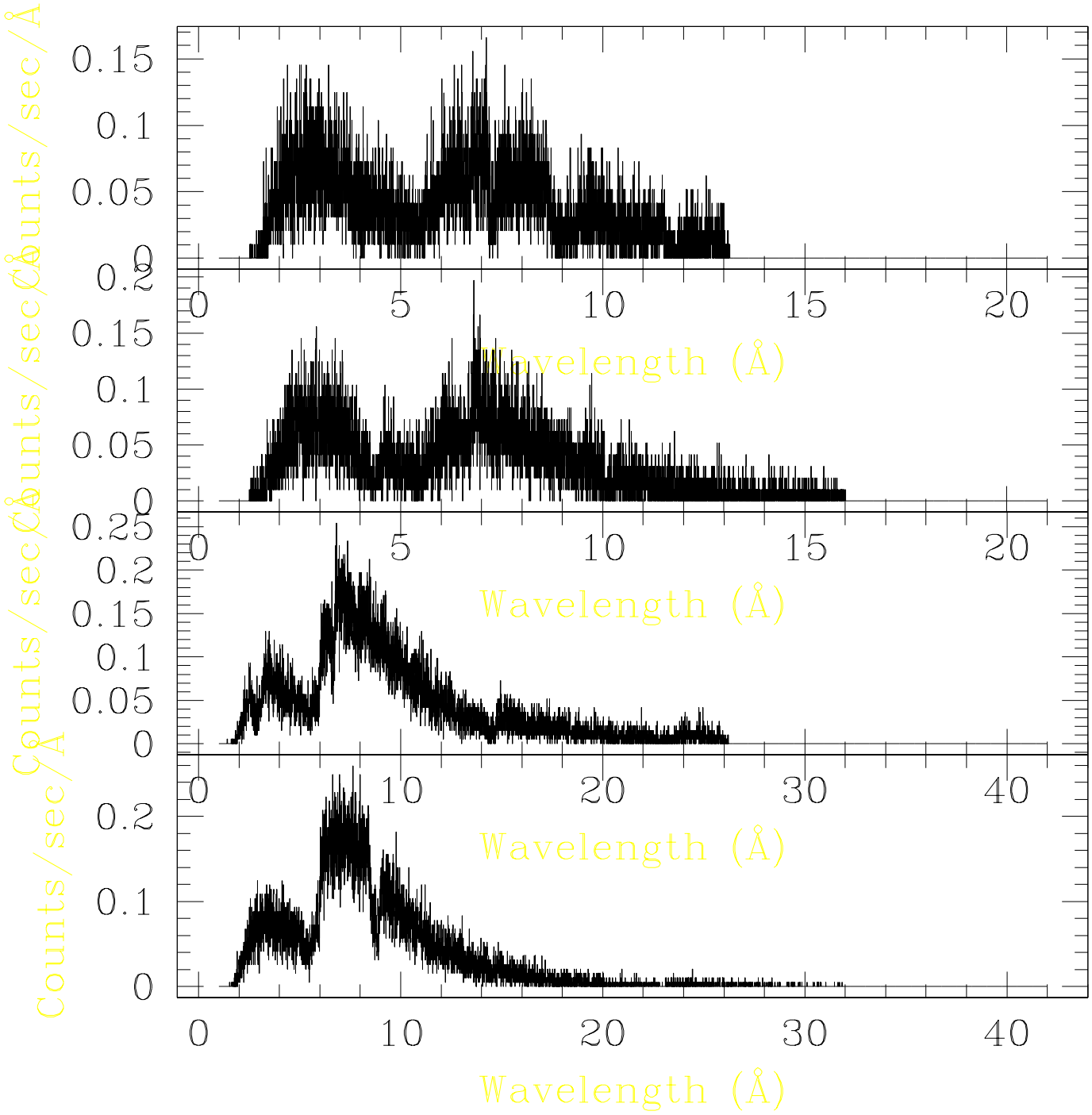
```
sherpa> SPLIT GAP 0.04
sherpa> D ALL XLABEL ""
sherpa> D 4 XLABEL "Wavelenght(\AA)"
sherpa> D 4 XLABEL SIZE 1.5
sherpa> D ALL TICKVALS 1.01
sherpa> D ALL YLABEL ""
sherpa> D 3 LABEL -2.2 0.1 "Counts/cm^2/\AA"
sherpa> L 1 ANGLE 90
sherpa> L 1 YELLOW
sherpa> L 1 SIZE 1.5
sherpa> TITLE "3c273 HEG/MEG -1/+1 orders simultaneously"
sherpa> D 1 LABEL 12 0.15 "HEG-1"
sherpa> L 1 RED
sherpa> D 2 LABEL 12 0.15 "HEG+1"
sherpa> L 1 RED
sherpa> D 3 LABEL 26 0.2 "MEG-1"
sherpa> L 2 RED
sherpa> D 4 LABEL 26 0.2 "MEG+1"
sherpa> L 1 RED
sherpa> REDRAW
```

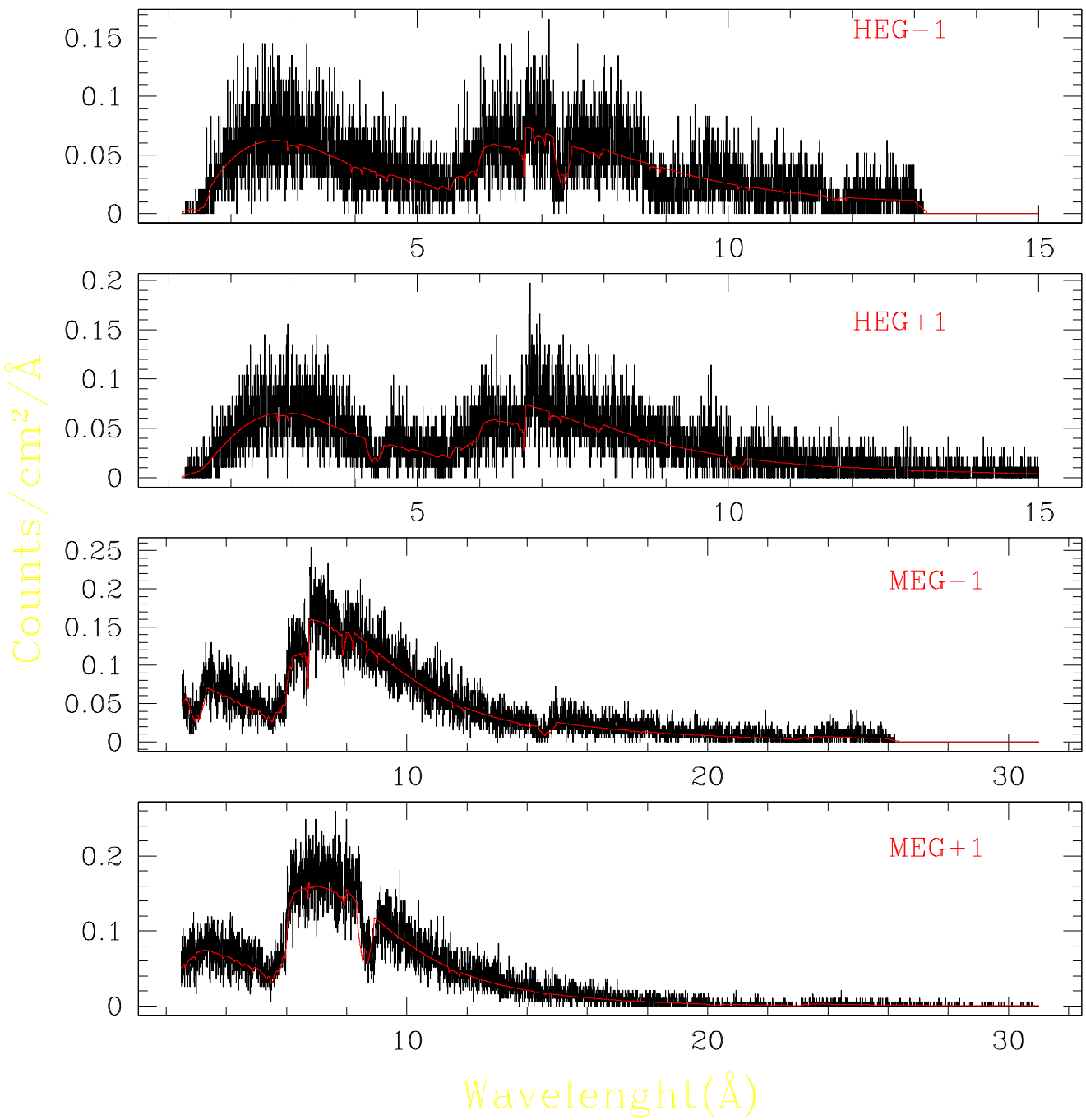
Note that `REDRAW` must be issued in *Sherpa* to view the effects of one or more previously issued *ChIPS* commands.

 Figure 3 shows the resulting plot. Finally, this figure may be saved as a PostScript file, and then printed:

```
sherpa> PRINT POSTFILE session1.ps
sherpa> $lpr session1.ps
```

Examining Fit Results


Counts/sec/Å
0.15
0.1
0.05
0
0.2
0.15
0.1
0.05
0
0.25
0.2
0.15
0.1
0.05
0
0.2
0.1
0
0
5
10
15
20
0
10
20
30
40
Wavelength (Å)


3c273 HEG/MEG −1/+1 orders simultaneously
HEG−1
HEG+1
MEG−1
MEG+1
Counts/cm²/Å
Wavelenght(Å)
0.15
0.1
0.05
0
5
10
15
0.2
0.25
20
30

CIAO 2.1 Science Threads

Extract Coadded and Grouped Nth-Order Source & Background Spectra and ARFs (4 April 2001)

Return to Threads Page.

For illustration, this thread utilizes data from the ObsID 459 dataset (HETG/ACIS-S, 3C 273), which was downloaded from the Archive. In this example, all level=1 and level=2 data files were downloaded.

This thread may be run on ACIS/HETG, ACIS/LETG, or HRC/LETG observational data.

Contents:

The goal of this thread is to generate and group a coadded positive and negative order grating spectrum (for both source and background) and the associated ARF.

A. Get Started

Before beginning this thread, you need to download the script add_grating_orders. You also need to have previously built positive and negative grating ARFs following the threads for building ACIS/HETG, ACIS/LETG and HRC/LETG ARFs.

B. Run the add_grating_orders Script

This script runs the following tools in order:

A. dmtype2split: to split the PHA2 spectrum into two temporary positive and negative order spectra with PHA1-like format.
B. dmtcalc: to rename columns in the two temporary PHA1 files, and to build two temporary single-order ARFs with renamed columns.

C. dmpaste: to merge the two temporary PHA1 spectra and ARFs.
D. dmtcalc: to add together positive and negative spectra and ARFs.
E. dmcopy: to build a final coadded spectrum with a PHA1-like format, and a coadded ARF.
F. dmgroup: to group the coadded positive and negative order spectrum (if needed).

(a detailed thread on how to perform the above steps manually will be posted soon)

Executing the script without any input files will produce this:

```
unix% add_grating_orders

This script adds together positive and negative order source and background ACI

Usage:

add_grating_orders pha2 order garm garf_negative garf_positive group_type group_

Where:
pha2: is the type 2 pha file output from tgextract
order: is the order of the grating spectra to extract
arm: is the grating arm to extract the spectrum from (HEG, MEG or LEG)
garf_negative/positive: are the grating ARF for negative and positive orders fo
group_type: is the grouping type (NONE|BIN|SNR|NUM_BINS|NUM_CTS|ADAPTIVE; see t
group_spec: is the grouping specification (see the dmgroup documentation for he
root: is the root name for the output files

EXAMPLES:

add_grating_orders acis_pha2.fits 1 HEG HEG_-1_garf.fits HEG_1_garf.fits BIN 10

unix%
```

This header shows what files are necessary to run the script. The command-line parameters `garm` and `group_type` are NOT case sensitive. To learn more about the several options for `group_type`, please see ahelp dmgroup (NB: the "ADAPTIVE" grouping may take a long time on large PHA files).

In this example, we build coadded 1st-order source and background MEG spectra and a 1st-order ARF. Then the source spectrum is grouped by a factor of 10:

```
unix% add_grating_orders acism00459N000_pha2.fits 1 MEG acism00459N000MEG_-1_ga

Input pha2 file is: acism00459N000_pha2.fits
Using MEG grating ARF order 1: acism00459N000MEG_-1_garf.fits
Using MEG grating ARF order 1: acism00459N000MEG_1_garf.fits
The root filename for the output file is: spec


MEG spectrum will be grouped by 10 = .050 Angstrom

Working on MEG spectra...

Splitting the pha2 file acism00459N000_pha2.fits in MEG +1/-1 orders...


Renaming columns of the pha1 files spec_MEG_m1.pha_tmp and spec_MEG_p1.pha_tmp.
```

```
Adding +1/-1 orders...


Filtering data files and building MEG order=1 spectrum spec_MEG_1.pha

Clobbering old output file: spec_MEG_1.pha

Working on MEG Effective Areas...


Adding +1/-1 orders...


Filtering data file and building MEG order=1 effective area spec_MEG_1.arf...
Clobbering old output file: spec_MEG_1.arf

Grouping MEG order=1 pha spectrum...

 WARNING: unable to read tlmin, tlmax.

 WARNING: unable to read tdbin.

The following errors occurred:
# 28021: dmgroup:
 WARNING: The maximum channel given in the binning specification ( 8192.000000
 given bin size ( 10.000000 ) and number of input file rows ( 8192 ); resetting
 channel to 8183.000000.

Cleaning up...
Done.

The coadded, grouped output spectrum is named spec_MEG_1_BIN10.pha
The coadded Effective Area is named spec_MEG_1.arf

You may now want to read both the MEG order=1 spectrum and effective area (spec_
The commands in sherpa are:

data spec_MEG_1_BIN10.pha
rsp[arfname](,spec_MEG_1.arf,)
instrument = arfname

unix%
```

The dmgroup errors can be ignored.

The thread is now complete. The coadded source and background 1st-order spectra are both contained in the PHA1-like file named "spec_MEG_1_BIN10.pha", while the coadded +/- 1st order ARF is named "spec_MEG_1.arf". To fit this data in *Sherpa*, note the instructions at the bottom of the of script's output. Then you can proceed to the Fitting Spectral Data in *Sherpa* thread.

Updates
05 March 2001 -- original version; identical to CIAO 2.0 version.
22 March 2001 -- modified date format
04 April 2001 -- added links back to Threads page