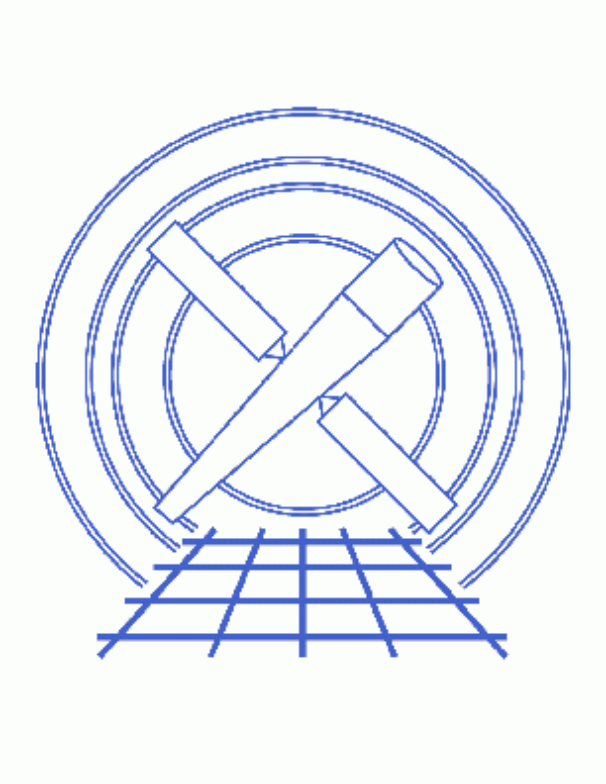
Examining Grating Spectra and Regions: PHA2 files



CIAO 3.4 Science Threads

Table of Contents

Examining Grating Spectra and Regions: PHA2 files

CIAO 3.4 Science Threads

Overview

Last Update: 1 Dec 2006 − updated for CIAO 3.4: ChIPS and Sherpa versions

Synopsis:

An overview of displaying grating data Type II PHA files and the source and background extraction regions.

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.

What is a PHA2 File?

A Type II PHA file is a standard FITS format in which each row contains array columns. The PHA file is a product of standard data processing and is identified by the `pha2.fits` extension; note that the "2" in the filename refers to the fact that it is a level=2 data product, ***not*** that it is a Type II file. In the case that the user has to manually reprocess an event file (e.g. when applying an updated order sorting table), the PHA2 spectrum file is obtained from the level 2 event file by `tgextract`; see the Obtain Grating Spectra from HETG/ACIS−S Data for an example of this.

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                    X sky coord of source
    6     Y                    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 colum
    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
```

There are two columns that are especially relevant when doing analysis:

- `TG_M` indicates the order of the spectrum (+/− 1, +/−2, +/− 3)
- `TG_PART` indicates the spectral component / grating arm (1 = HEG, 2 = MEG, 3 = LEG)

Examining the Files with Prism

Sample ObsIDs used: 459 (HETG/ACIS−S 3C 273); 460 (LETG/HRC−S, 3C 273); 1198 (LETG/ACIS−S, 3C 273); 1800 (LETG/HRC−I, PKS2155−304)

File types needed: evt2; pha2

ACIS−S HETG/LETG Observations

We can use `prism` to examine the PHA2 file for ObsID 459:

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

which will give you something like Figure 1  . In this example, there are twelve rows − all the +/− orders for both HEG and MEG − for the observation. 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.

An ACIS−S/LETG observation (ObsID 1198) looks similar in *Prism*  but only contains 6 rows (+/− orders for the LEG).

Each vector column can be viewed as follows:

```
unix% prism acisf00459N002_pha2.fits &

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

Figure 3  shows the (partial) result of the expansion of column `BIN_LO`.

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 hrcf00460N002_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` and `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 Figure 5  in the example of ObsID 1800.

Displaying the Spectrum

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 acisf00459N002_pha2.fits &

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

As clearly visible from the output in Figure 6  , the plot is good for a general overview only (are there enough counts, are there striking features, etc.). At the same time as the plot was created, a *ChIPS* session was launched by *Prism* in an xterm window. This window can be used to customize the plot; see the Introduction to *ChIPS* thread for more information on plotting with *ChIPS*.

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):

```
unix% dmtype2split "acisf00459N002_pha2.fits[#row=4]" acisf00459N002_heg_p1.pha
```

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

```
unix% chips

Welcome to ChIPS, version CIAO 3.4
Copyright (C) 1999−2003, Smithsonian Astrophysical Observatory

chips> plot "acisf00459N002_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"
```

These commands produce the plot shown in Figure 7  .

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. To produce the plot above using S−Lang syntax:

```
chips> clear
chips> pha2 = readbintab("acisf00459N002_pha2.fits")
chips> print(pha2)
_filename        =  acisf00459N002_pha2.fits
_path            =  /data/ciao/threads/
_filter          =  NULL
_filetype        =  4
```

```
_header          =  String_Type[190]
_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> curve(bin_lo_heg_p1,counts_heg_p1)
0
chips> symbol none
chips> step
chips> xlabel "Wavelength (\AA)"
chips> ylabel Counts
chips> title "ACIS+HEG order=+1"
```

Note that S−Lang follows C conventions by numbering array indexes from 0, rather than 1. Therefore the 4th row (HEG, +1 order) in the PHA2 file 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,*]`)".

With Sherpa

Sherpa can also be used to plot a PHA2 spectrum. In this case, the spectra do not need to be split first; *Sherpa* reads all the rows and allows you to specify individual ones for plotting or fitting purposes:

```
unix% sherpa

-------------------------------------------------------
Welcome to Sherpa: CXC's Modeling and Fitting Program
-------------------------------------------------------
Version: CIAO 3.4

Type AHELP SHERPA for overview.
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 $ASCDS_WORK_PATH in your 'ciao' setup script.

    Abundances set to Anders & Grevesse

sherpa> data acisf00459N002_pha2.fits
The inferred file type is PHA Type II.  If this is not what you want, please
specify the type explicitly in the data command.
Warning: could not find SYS_ERR column
WARNING: statistical errors specified in the PHA file.
         These are currently IGNORED.  To use them, type:
```

```
         READ ERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
WARNING: backgrounds UP and DOWN are being read from this file,
         and are being combined into a single background dataset.
WARNING: multiple datasets have been input.
         The next available dataset number is 13.

sherpa> sherpa.dataplot.curvestyle="histo"
sherpa> sherpa.dataplot.symbolstyle="none"
sherpa> analysis channel
sherpa> lp 2 data 3 data 4
```

Figure 8 shows the plot of the HEG −1 order (row 3, upper drawing area) and +1 order (row 4, lower drawing area) that is created. To change the axes to CHANNEL vs. COUNTS:

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

These commands alter the plots to look like Figure 9. *ChIPS* commands (within *Sherpa*) can be used at this point to customize the plot.

```
sherpa> exit
Goodbye.
```

Displaying the Extraction Regions

Each pha2 file has a second block, named REGION, which stores the regions used by tgextract to extract the source and background spectra. There are three regions associated with each order: source, upper background, and lower background. For an ACIS/HETG observation, this gives 36 regions: 12 spectral components (+/− 3, +/− 2, and +/− 1 for HEG and MEG) times 3 regions apiece (source and two backgrounds).

To look at the columns of a REGION block:

```
unix% dmlist "acisf00459N002_pha2.fits[REGION]" cols

--------------------------------------------------------------------------------
Columns for Table Block REGION
--------------------------------------------------------------------------------

ColNo  Name                Unit        Type            Range
   1   SPEC_NUM                         Int2          1:32767             Spectrum number
   2   ROWID                            String[64]                        Source or a background
   3   SHAPE                            String[16]                        Shape of region
   4   TG_LAM              angstrom     Real4         0:      400.0       Dispersion coordinate
   5   TG_D                degrees      Real4         −2.0:       2.0     Cross−dispersion coordi
   6   R[2]                (angstrom , degrees) Real4(2)      −Inf:+Inf    Raduis vector for SHAPE
   7   ROTANG              degrees      Real4         −360.0:     360.0   Rotation angle for SHAP
   8   TG_PART                          Int2          0:9                 Grating part index (HEG
   9   TG_SRCID                         Int2          1:32767             Source identification n
  10   TG_M                             Int2          −62:62              Diffraction order
  11   COMPONENT                        Int2          −                   Component number
```

ds9 cannot display these regions as they are written in the pha2 file. In order to view them, we need to rename the (TG_LAM,TG_D) columns to (X,Y) so that ds9 knows how to interpret them. We will also need to create images in (TG_LAM,TG_D) coordinates, on which we can display the regions.

The following dmcopy commands create image and region files for the 1st and 3rd orders of the MEG arm:

```
unix% dmcopy \
      "acisf00459N002_evt2.fits[bin tg_lam=0:30:0.08,tg_d=-0.01:0.01:0.00008][tg_m=-1,1,tg_part=2]" \
      459_order1.fits

unix% dmcopy \
      "acisf00459N002_evt2.fits[bin tg_lam=0:15:0.08,tg_d=-0.01:0.01:0.00008][tg_m=-3,3,tg_part=2]" \
      459_order3.fits

unix% dmcopy \
      "acisf00459N002_pha2.fits[region][tg_m=1,tg_part=2][cols x=tg_lam,y=tg_d,*]" \
      region_order1.fits

unix% dmcopy \
      "acisf00459N002_pha2.fits[region][tg_m=3,tg_part=2][cols x=tg_lam,y=tg_d,*]" \
      region_order3.fits
```

In creating the images, the filter includes + and − orders to obtain more events in the image. Since the regions are the same for +/− orders, it is only necessary to copy one (the + orders were used here). The image limits are typical for ACIS/HETG observations, but will need to be adjusted for other configurations.

To display the event files with the regions ovelaid:

```
unix% ds9 -tile 459_order1.fits -region region_order1.fits -cmap a\
      459_order3.fits -region region_order3.fits -cmap a
```

which produces Figure 10  . We can see that all events are contained within at least one extraction region.

History

01 Jun 2004	reviewed for CIAO 3.2: no changes
06 Dec 2005	updated for CIAO 3.3: version numbers
01 Dec 2006	updated for CIAO 3.4: ChIPS and Sherpa versions

URL: http://cxc.harvard.edu/ciao/threads/examinepha2/ Last modified: 1 Dec 2006

Image 1: Prism view of ACIS−S HETG dataset

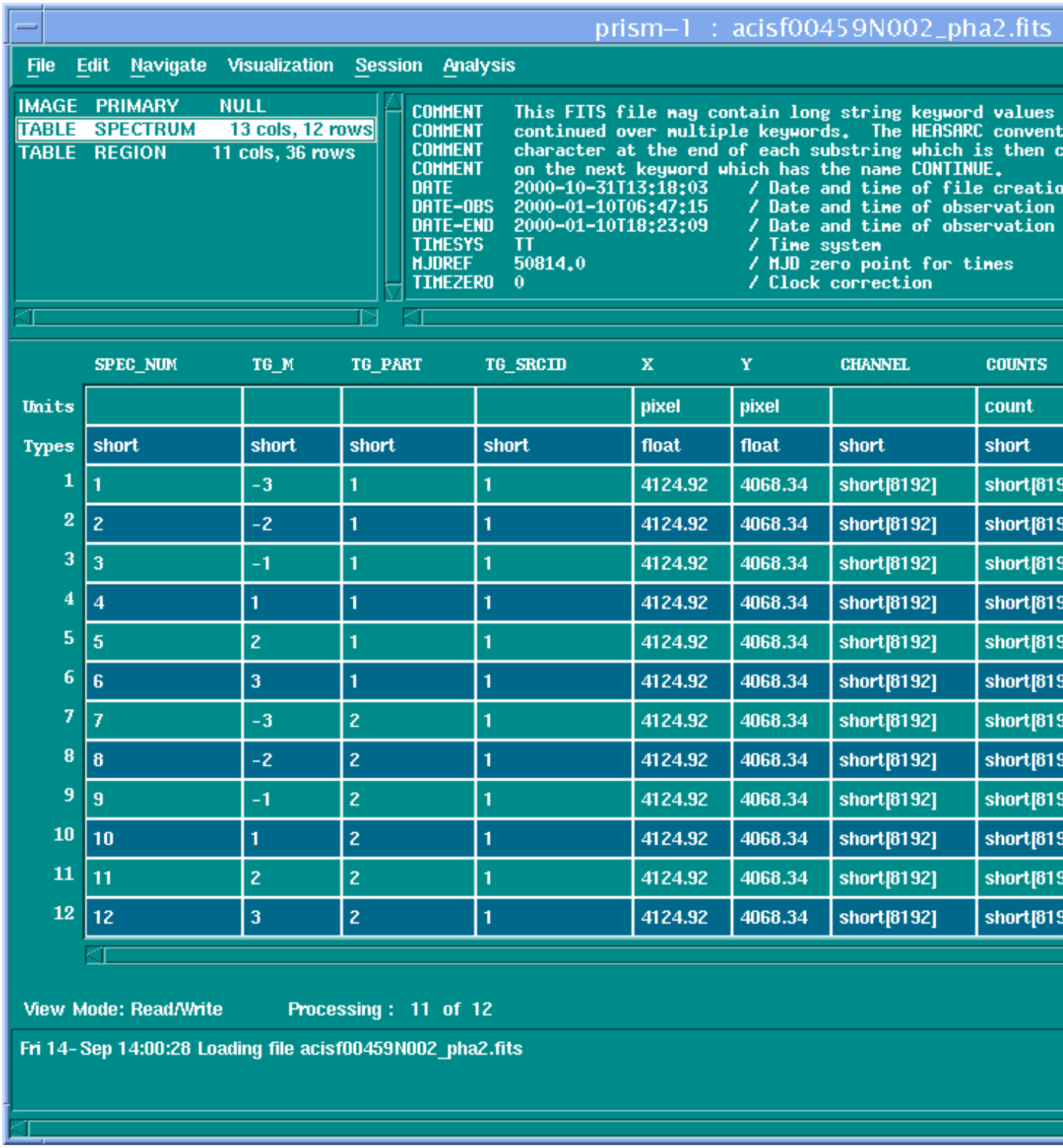


Image 2: Prism view of ACIS−S LETG dataset

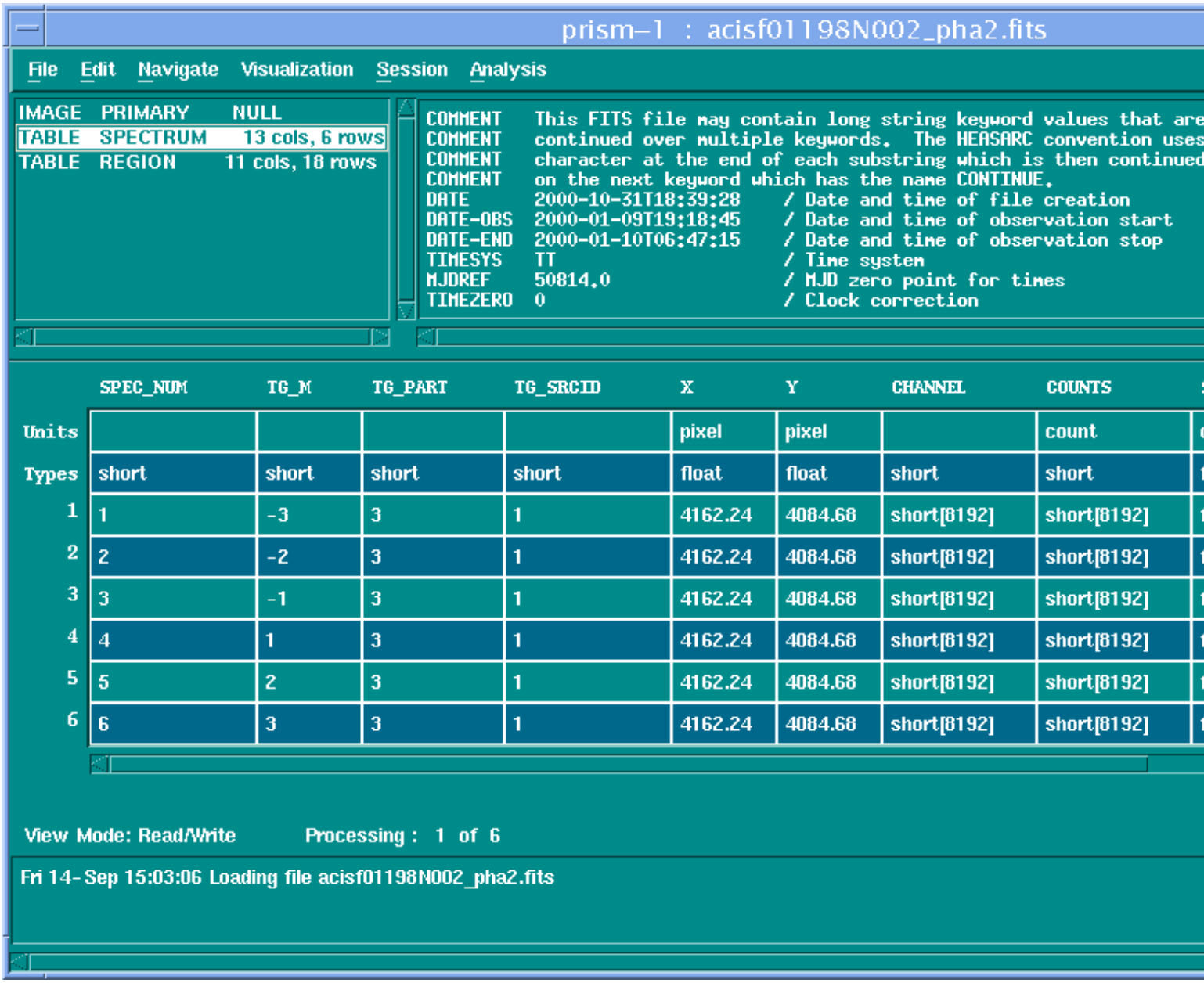


Image 3: Expanded view of BIN_LO column

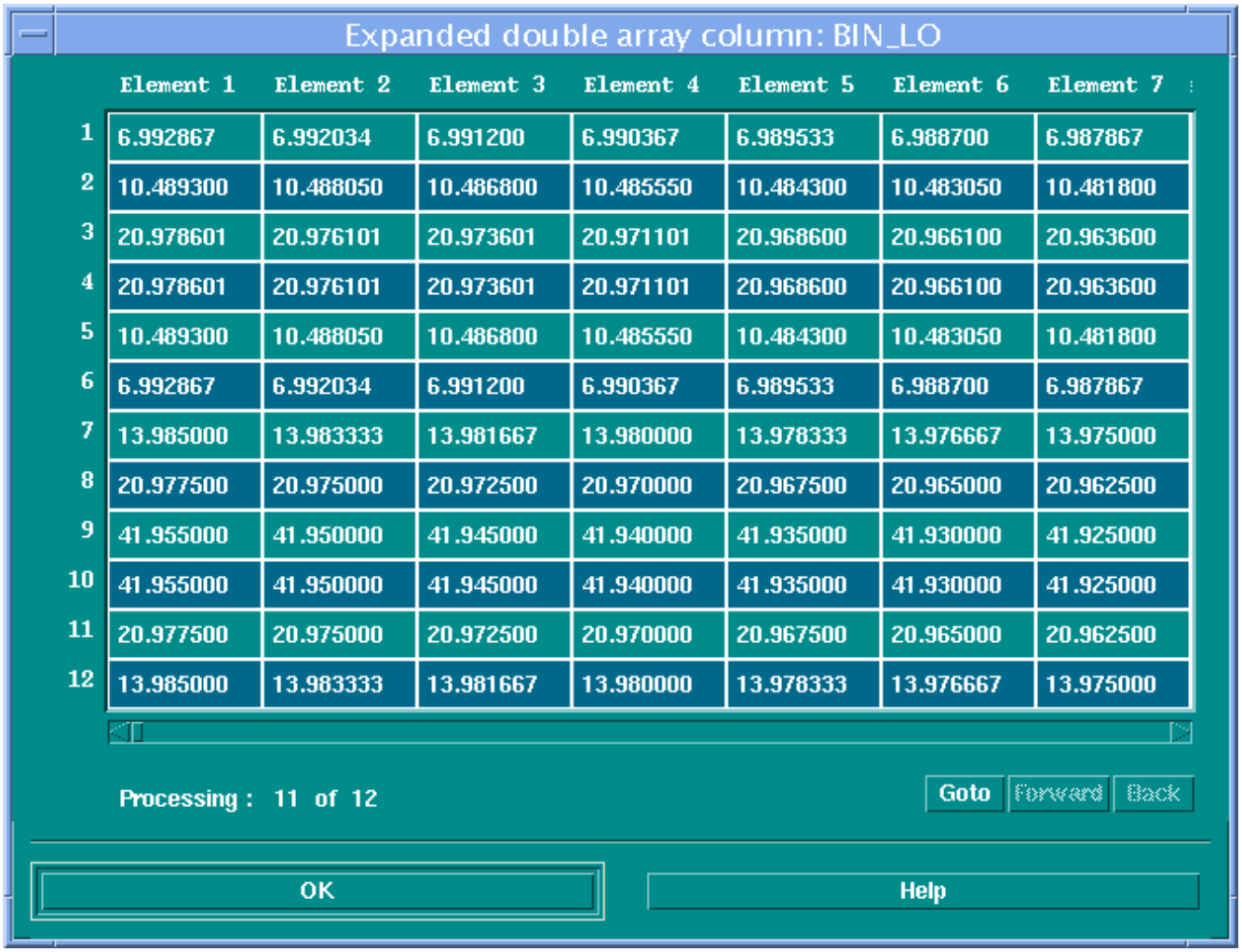


Image 4: Prism view of HRC−S LETG dataset

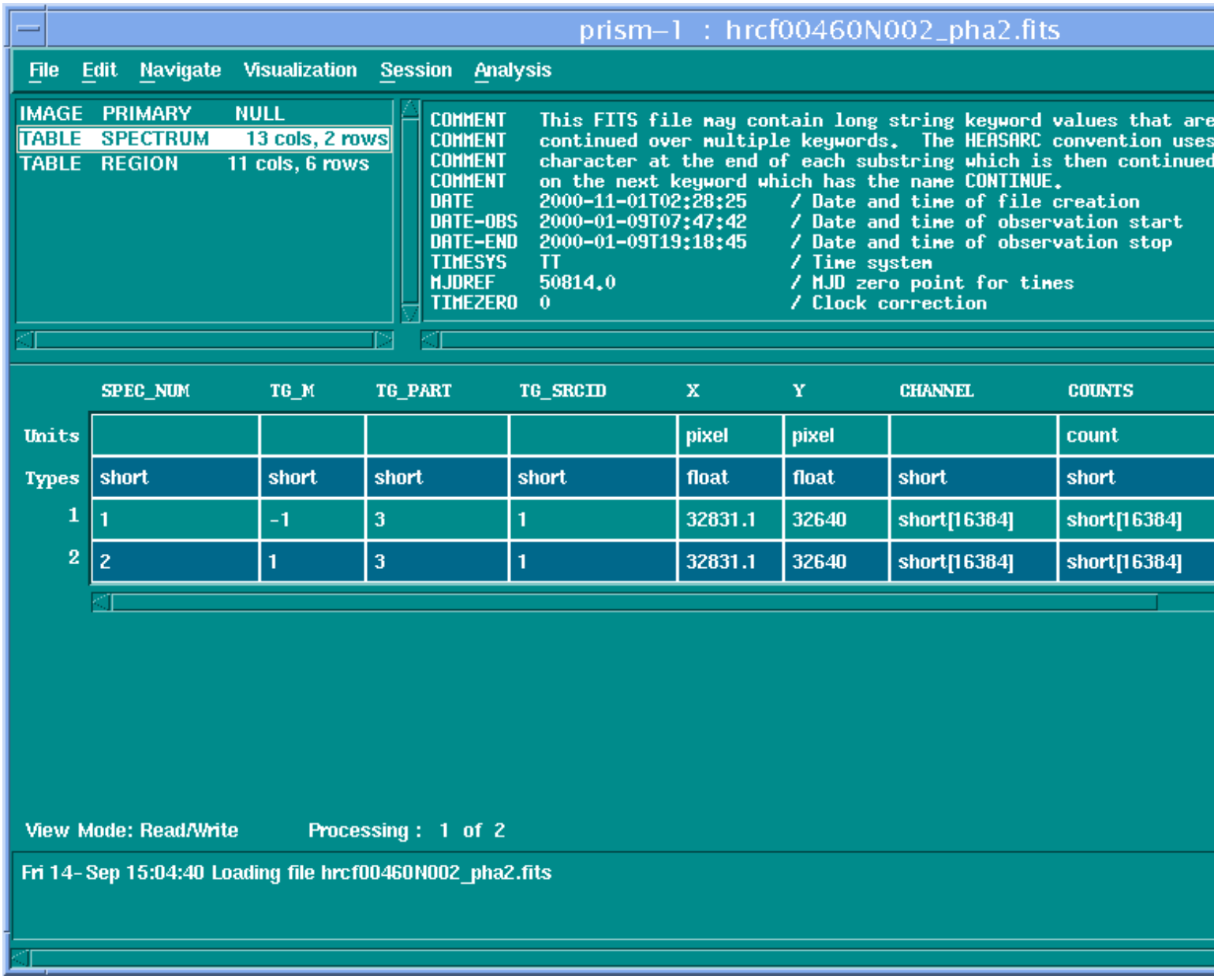


Image 5: Prism view of HRC−I LETG dataset

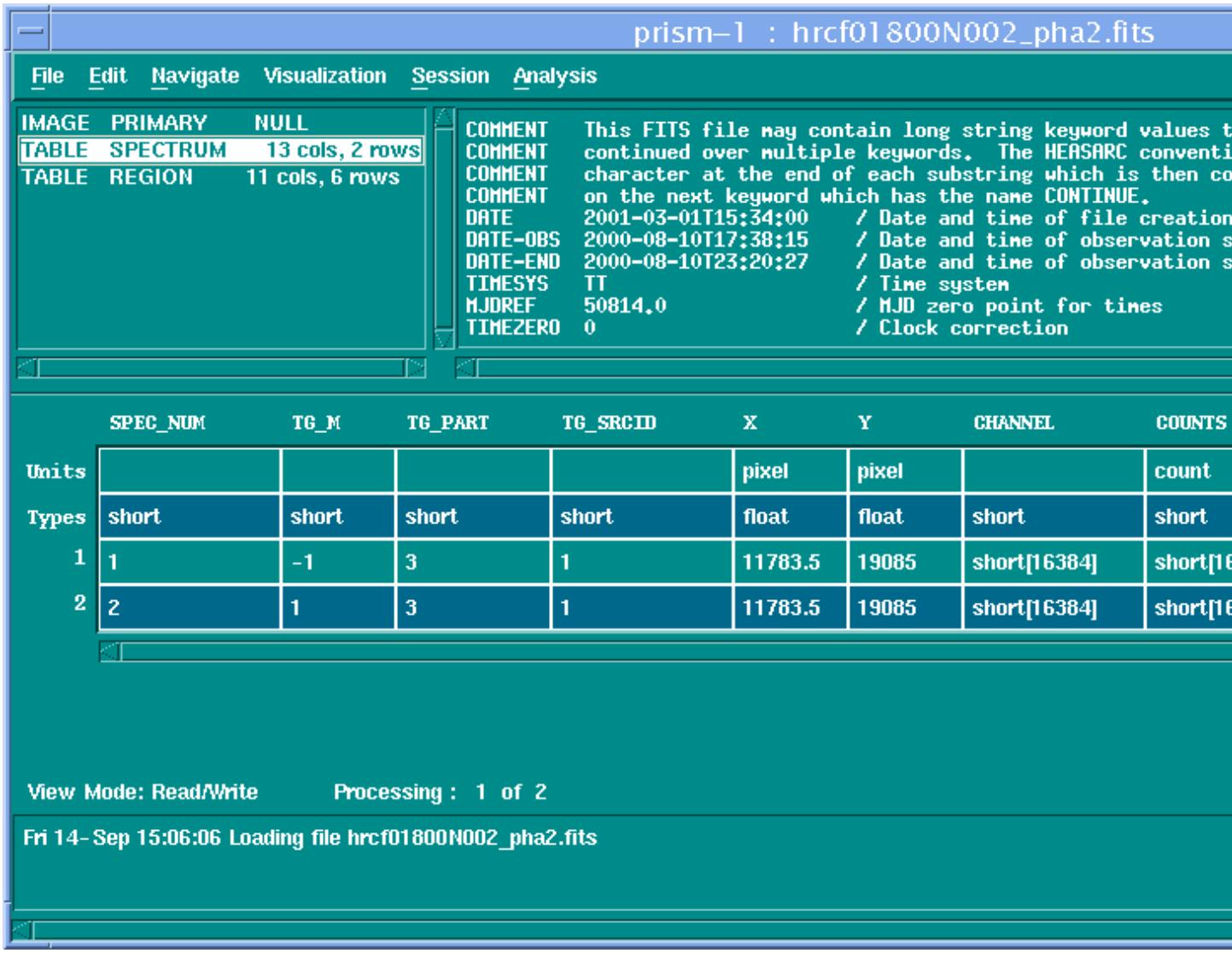


Image 6: View of order +1 spectrum with Prism

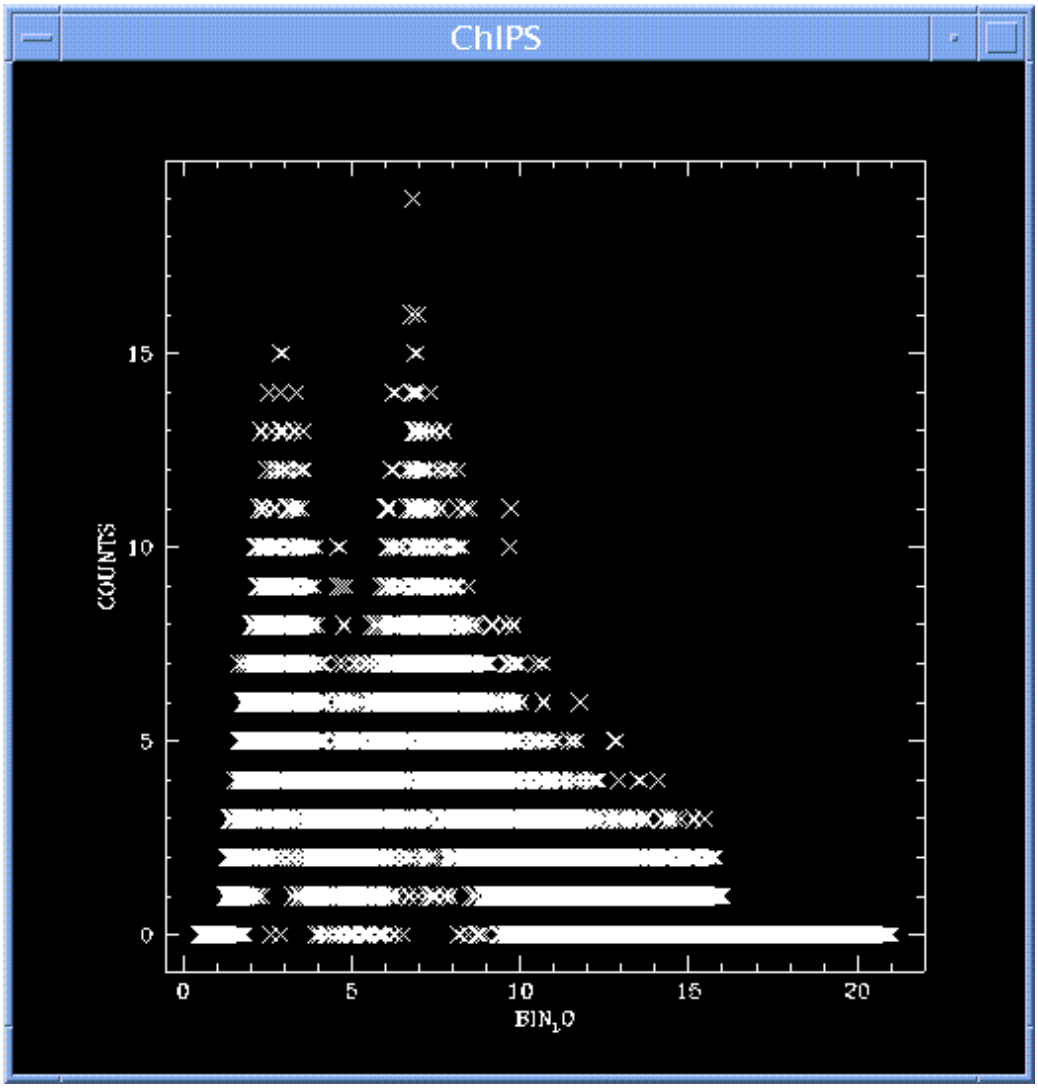


Image 7: View of order +1 spectrum with ChIPS

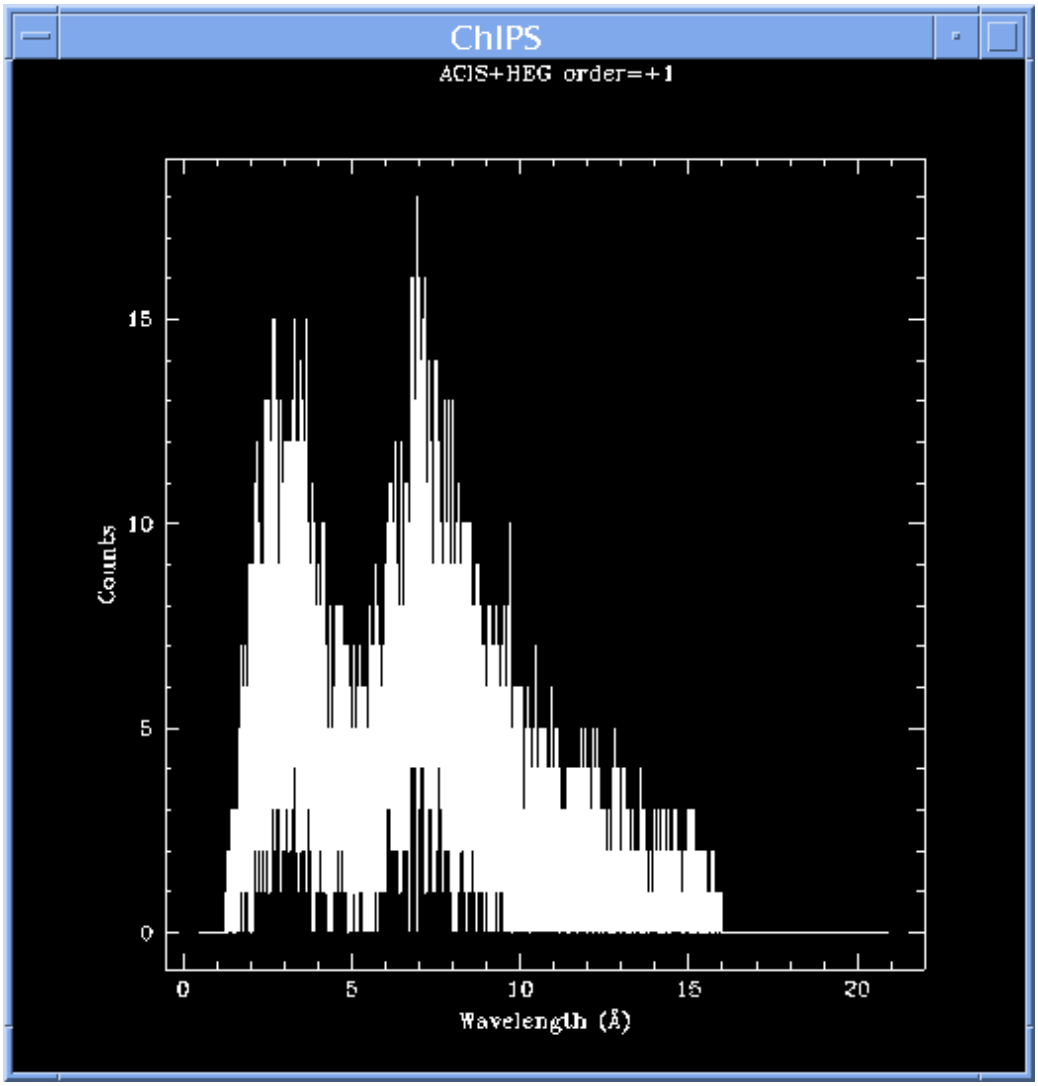


Image 8: View of order +/−1 spectrum with Sherpa: count rate

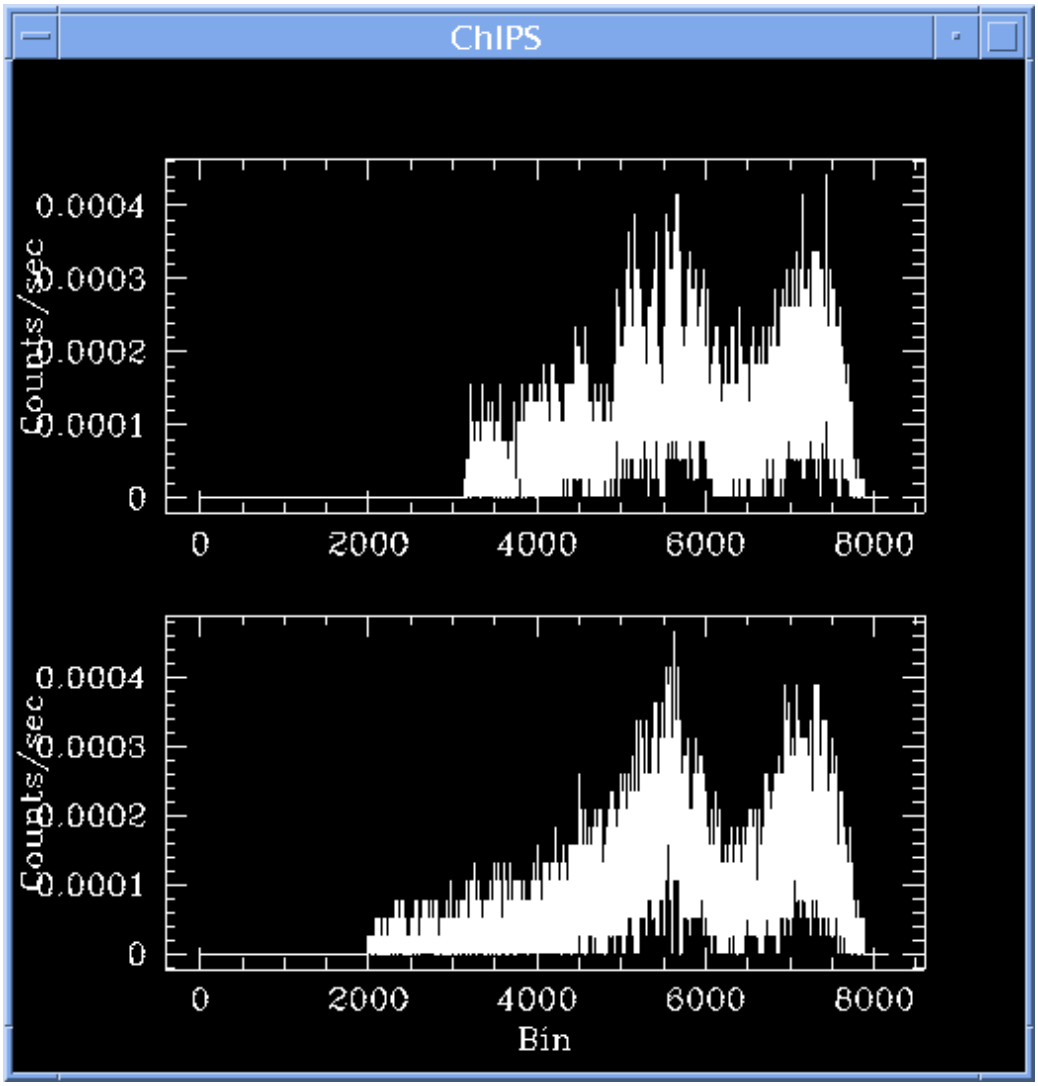


Image 9: View of order +/−1 spectrum with Sherpa: counts

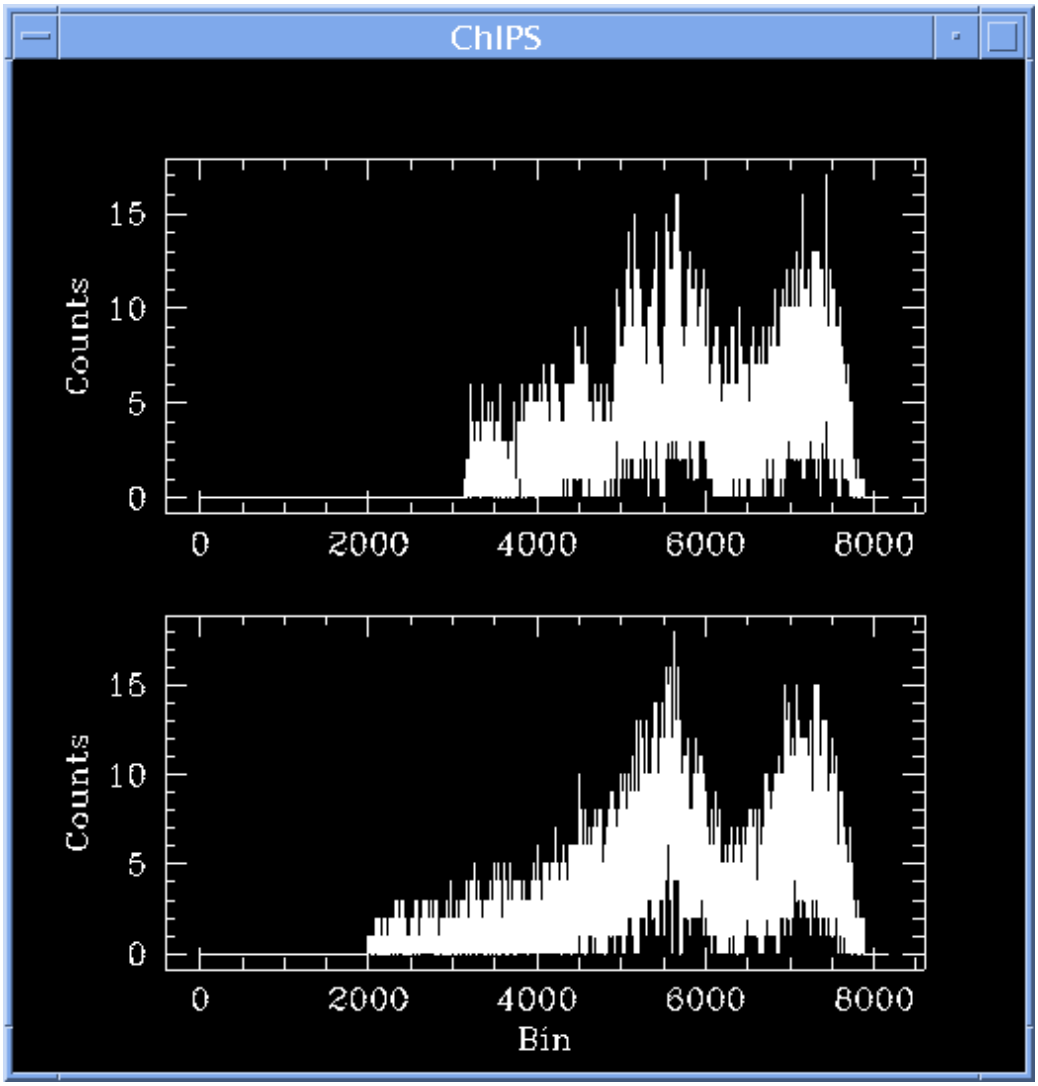


Image 10: Data with source and background regions overlaid

The first order (left) and third (right) order images with extraction regions overlaid.