Independent Background Responses

Sherpa Threads (CIAO 3.4)

Table of Contents

Independent Background Responses

Sherpa Threads

Overview

Last Update: 1 Dec 2006 – reviewed for CIAO 3.4: no changes

Synopsis:

If you would like to fit a background–subtracted source spectrum using a common RMF and ARF for source and background, simply read the source spectrum fits file into *Sherpa*, subtract the background, and fit it. To fit source and background spectra simultaneously with proper and distinct RMFs and ARFs, load the source and background as different datasets. This thread illustrates both cases.

While the sample datafiles used in this thread are available as sherpa.tar.gz, note also that they are generated with the Extract ACIS Spectra for Pointlike Sources and Make RMFs and ARFs thread.

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

Getting Started

Please follow the "Sherpa Threads: Getting Started" thread.

Statistical Issues: Background Subtraction

A typical dataset may contain multiple spectra, one of which contains contributions from the source of interest and the background, and one or more others which contain background counts alone. (The background itself may contain contributions from the cosmic X–ray background, the particle background, etc., but we'll ignore this complication.) The proper way to treat background data is to model them!

However, many X–ray astronomers *subtract* background data from the raw data.

Why should one *not subtract* background?

- It reduces the amount of statistical information in the analysis – the final fit parameter values will be a less accurate estimate of the true values.
- The background subtracted data are not Poisson–distributed – one cannot fit them with the Poisson likelihood (or the Cash statistic), even in the low–count limit. For example, subtracting a background can give *negative counts*; this is definitely not Poissonian!
- Fluctuations (particularly in the vicinity of localized features) can adversely affect analysis.

Fit a Background–Subtracted Source

Reading FITS data for source and background

By default, psextract updates the header keywords `BACKFILE`, `RESPFILE`, and `ANCRFILE` in the source spectrum file:

```
unix% dmkeypar 3c273.pi BACKFILE echo+
3c273_bg.pi

unix% dmkeypar 3c273.pi RESPFILE echo+
3c273.rmf

unix% dmkeypar 3c273.pi ANCRFILE echo+
3c273.arf
```

The `RESPFILE` and `ANCRFILE` keywords are not updated in the ungrouped background file, however, and should be manually added with dmhedit, as shown in the Update File Headers section of the psextract thread.

On account of these keywords, *Sherpa* automatically reads in the (ungrouped) background spectrum "3c273_bg.pi" and the source response matrices "3c273.rmf" and "3c273.arf" when the source spectrum is read:

```
sherpa> DATA 3c273.pi
The inferred file type is PHA.  If this is not what you want, please
specify the type explicitly in the data command.
WARNING: statistical errors specified in the PHA file.
         These are currently IGNORED.  To use them, type:
         READ ERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
Background data are being input from:
  3c273_bg.pi
WARNING: statistical errors specified in the PHA file.
         These are currently IGNORED.  To use them, type:
         READ BERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
RMF is being input from:
  3c273.rmf
ARF is being input from:
  3c273.arf
```

If *Sherpa* does not automatically read in the background data, then it can be input as follows:

```
sherpa> BACK 3c273_bg.pi
The inferred file type is PHA.  If this is not what you want, please
specify the type explicitly in the data command.
WARNING: statistical errors specified in the PHA file.
         These are currently IGNORED.  To use them, type:
         READ BERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
```

Subtracting the background

The user can then subtract the background from the source spectrum and fit the background–subtracted source spectrum using response matrices for the source only. The following commands may be used as an example script for subtracting the background and then visualizing the 0.3–3 keV background–subtracted source data in *Sherpa*:

```
sherpa> SUBTRACT
sherpa> LP DATA
sherpa> IGNORE SOURCE ENERGY :0.3,3.:
sherpa> LP DATA
```

```
sherpa> LOG X
sherpa> LOG Y
sherpa> LIM Y 1.e-3 0.2
sherpa> LIM X 0.2 3.5
sherpa> REDRAW
```

Which produces this plot 📷.

Defining the source instrument response

Since the input source spectrum referenced the instrument response files in its header, an instrument model was automatically defined:

```
sherpa> SHOW INSTRUMENT
Instrument 1: rsp1d[AutoReadResponse]
    Param   Type  Value
    -----   ----  -----
 1   rmf string: "3c273.rmf" (N_E=1090,N_PHA=1024)
 2   arf string: "3c273.arf" (N_E=1090)
```

The command SHOW reports further information about the currently established models:

```
sherpa> SHOW
(cut)

-----------------------------
Defined analysis model stacks:
-----------------------------

instrument source 1 = AutoReadResponse
instrument back 1 = AutoReadResponse

(cut)
```

Note that the instrument model was automatically set up for both the source and background. However, unless a background model is defined for fitting, using the BACKGROUND command, then the background data will be ignored.

If you manually read in a background file after reading the source data, the background model stack was reset. To set it to be identical to the source model stack:

```
sherpa> INSTRUMENT BACK 1 = AutoReadResponse
```

If *Sherpa* does not automatically read in the response files and define an instrument model, then it can be set up as follows:

```
sherpa> INSTRUMENT SOURCE 1 = RSP[mydetector]("3c273.rmf", "3c273.arf")
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.

sherpa> SHOW INSTRUMENT
Instrument 1: rsp1d[mydetector]
    Param   Type  Value
    -----   ----  -----
 1   rmf string: "3c273.rmf" (N_E=1090,N_PHA=1024)
 2   arf string: "3c273.arf" (N_E=1090)

sherpa> INSTRUMENT BACK 1 = mydetector
```

Defining and fitting a source model

The following commands may be used as an example script for fitting the background–subtracted data with a source model consisting of an absorbed power law:

```
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> XSWABS[abs](0.1)
sherpa> POW[pl](2,1,0.01:0:1e10)
sherpa> SOURCE = abs * pl
sherpa> FIT
 LVMQT: V2.0
 LVMQT: initial statistic value = 573523
 LVMQT: final statistic value = 14.6037 at iteration 9
          abs.nH  0.00875097  10^22/cm^2
           pl.gamma  1.70028
           pl.ampl  0.000166814

sherpa> LP 2 FIT RESIDUALS
sherpa> D 1:2 LOG X
sherpa> D 1 LOG Y
sherpa> D 1 LIM Y 1.e-3 0.2
sherpa> D 1:2 LIM X 0.2 3.5
sherpa> D 1 TICKVALS OFF
sherpa> D 1 XLABEL
sherpa> REDRAW
```

The results are shown here 📷.

Simultaneously Fit Source and Background with the Same Responses

Instead of subtracting the background, the user may choose to simultaneously fit the source and background spectra, using the same (source) RMF and ARF.

Reading FITS data for source and background

Again, *Sherpa* automatically reads in the (ungrouped) background spectrum and the source response matrices when the source spectrum is read:

```
sherpa> ERASE ALL
sherpa> DATA 3c273.pi
The inferred file type is PHA.  If this is not what you want, please
specify the type explicitly in the data command.
WARNING: statistical errors specified in the PHA file.
        These are currently IGNORED.  To use them, type:
        READ ERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
Background data are being input from:
  3c273_bg.pi
WARNING: statistical errors specified in the PHA file.
        These are currently IGNORED.  To use them, type:
        READ BERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
RMF is being input from:
  3c273.rmf
ARF is being input from:
  3c273.arf
```

If *Sherpa* does not automatically read in the background data, then it can be input as follows:

Defining and fitting a source model

```
sherpa> BACK 3c273_bg.pi
The inferred file type is PHA.  If this is not what you want, please
specify the type explicitly in the data command.
WARNING: statistical errors specified in the PHA file.
        These are currently IGNORED.  To use them, type:
        READ BERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
```

Defining the source and background instrument response

Since the input source spectrum referenced the instrument response files in its headers, an instrument model was automatically defined:

```
sherpa> SHOW INSTRUMENT
Instrument 1: rsp1d[AutoReadResponse]
    Param    Type  Value
    -----    ----  -----
 1    rmf string: "3c273.rmf" (N_E=1090,N_PHA=1024)
 2    arf string: "3c273.arf" (N_E=1090)
```

The instrument model was automatically set up for both the source and background. However, unless a background model is defined for fitting, using the BACKGROUND command, then the background data will be ignored. In order to fit the background simultaneously with the source, the BACKGROUND command will be issued in the next step.

If you manually read in a background file after reading the source data, the background model stack was reset. To set it to be identical to the source model stack:

```
sherpa> INSTRUMENT BACK 1 = AutoReadResponse
```

If *Sherpa* does not automatically read in the response files and define an instrument model, then it can be set up as follows:

```
sherpa> INSTRUMENT 1 = RSP[mydetector]("3c273.rmf", "3c273.arf")
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.

sherpa> SHOW INSTRUMENT
Instrument 1: rsp1d[mydetector]
    Param    Type  Value
    -----    ----  -----
 1    rmf string: "3c273.rmf" (N_E=1090,N_PHA=1024)
 2    arf string: "3c273.arf" (N_E=1090)

sherpa> INSTRUMENT BACK 1 = mydetector
```

```
sherpa> INSTRUMENT = RSP[mydetector]("3c273.rmf", "3c273.arf")
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.

sherpa> SHOW
.
.
----------------------------
Defined analysis model stacks:
----------------------------

instrument source 1 = mydetector
instrument back 1 = mydetector
.
.
```

Note that if no argument (e.g. SOURCE or BACKGROUND) is specified with the INSTRUMENT command, then the instrument model being defined will be used for both the source and background data.

Defining and fitting source and background models

Here we simultaneously fit the source and background with models consisting of an absorbed power law. The source and background models are defined using the SOURCE and BACKGROUND commands respectively.

```
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> XSWABS[abs](0.1)
sherpa> POW[plsrc](2,1,0.01:0:1e10)
sherpa> POW[plbkg](2,1,0.01:0:1e10)
sherpa> SOURCE = abs * plsrc
sherpa> BACKGROUND = abs * plbkg

sherpa> FIT
 LVMQT: V2.0
 LVMQT: initial statistic value = 5.446e+07
 LVMQT: final statistic value = 97.54 at iteration 14
          abs.nH  0.0173134  10^22/cm^2
        plsrc.gamma  1.9092
        plsrc.ampl  0.000173676
        plbkg.gamma  0.823776
        plbkg.ampl  8.91066e-07

sherpa> LP 2 FIT RESIDUALS
sherpa> D 1:2 LOG X
sherpa> D 1 LOG Y
sherpa> D 1 LIM Y 1.e-3 0.2
sherpa> D 1:2 LIM X 0.2 3.5
sherpa> D 1 TICKVALS OFF
sherpa> D 1 XLABEL
sherpa> REDRAW
```

The results are shown here 📷.

Simultaneously Fit Source and Background with Independent Responses

Instead of subtracting the background, the user may choose to simultaneously fit the source and background spectra, each with its own RMF and ARF.

Reading FITS data for source and background

In this thread, we wish to fit spectral data from the FITS (ungrouped) datafiles source.pi and back.pi. These datasets are input into *Sherpa* with the DATA command:

```
sherpa> ERASE ALL
sherpa> DATA source.pi
The inferred file type is PHA.  If this is not what you want, please
specify the type explicitly in the data command.
WARNING: statistical errors specified in the PHA file.
        These are currently IGNORED.  To use them, type:
        READ ERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
```

The background dataset is loaded similarly with the BACK command:

```
sherpa> BACK back.pi
The inferred file type is PHA.  If this is not what you want, please
```

```
specify the type explicitly in the data command.
WARNING: statistical errors specified in the PHA file.
        These are currently IGNORED.  To use them, type:
        READ BERRORS "<filename>[cols CHANNEL,STAT_ERR]" fitsbin
```

Now the dataset can be plotted:

```
sherpa> LPLOT 2 DATA BACK
```

The plot is visible here 📷. The data are plotted in bin (i.e. channel) space, since we have not yet defined the instrument response (RMFs and ARFs).

Defining source and background instrument responses

Here we establish a 1–D instrument model; they are named "r1" (source) and "r2" (background):

```
sherpa> PARAMPROMPT ON
Model parameter prompting is on
sherpa> RSP[r1]
r1.rmf parameter value ["none"] source.rmf
r1.arf parameter value ["none"] source.arf
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.

sherpa> RSP[r2]
r2.rmf parameter value ["none"] back.rmf
r2.arf parameter value ["none"] back.arf
The inferred file type is ARF.  If this is not what you want, please
specify the type explicitly in the data command.
```

It is also possible to establish a response model using *either* an RMF *or* an ARF file. The instrument help file has more information on this.

In order to convolve the input datasets with the instrument models that have been established, it must be defined as the INSTRUMENT model:

```
sherpa> INSTRUMENT SOURCE = r1
sherpa> INSTRUMENT BACK = r2
```

The current definition of *Sherpa*'s instrument model may be examined using SHOW:

```
sherpa> SHOW
.
.
-----------------------------
Defined analysis model stacks:
-----------------------------

instrument source 1 = r1
instrument back 1 = r2

------------------------------------
Defined instrument model components:
------------------------------------

rsp1d[r1]
    Param    Type  Value
    -----    ----  -----
 1    rmf string: "source.rmf" (N_E=1090,N_PHA=1024)
 2    arf string: "source.arf" (N_E=1090)

rsp1d[r2]
    Param    Type  Value
    -----    ----  -----
```

```
1    rmf string: "back.rmf" (N_E=1090,N_PHA=1024)
2    arf string: "back.arf" (N_E=1090)
```

The output shows that `rmf.fits` and `arf.fits` currently define the source instrument model.

Defining and fitting source and background models

We now need to define a <u>SOURCE</u> model for our source dataset and a <u>BACKGROUND</u> model for our background dataset. For these data, we find that absorbed blackbody spectra are appropriate. We need to define a model for the absorbing column and a blackbody for both the source and background:

```
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> XSWABS[a1]
sherpa> XSBBODY[b1]
sherpa> XSBBODY[b2]
```

We can then define the source model for the source and background with the <u>SOURCE</u> and <u>BACKGROUND</u> commands:

```
sherpa> SOURCE = a1*b1
sherpa> BACKGROUND = a1*b2
sherpa> IGNORE energy :0.3,10:
sherpa> LPLOT 2 FIT BACKFIT
```

Note that we also ignore the high and low energy regions of the spectrum, as they generally have lower quality data. The plot of the data with (default–value) fits produced by `LPLOT` is visible here 📷.

Clearly, the default values are not a very good fit to the data. We may choose to begin the fit now, using the `FIT` command, or we can refine the values of the fit by eye, prior to fitting. By varying parameter values, while plotting and replotting the data, we can help the fitting algorithm find the best minimum. By setting the parameters to values near what we expect, we also help avoid local minima in the parameter space. (Fitting spectra is something of an art; one generally gets better fits when they have *a priori* knowledge of the source.) We set the values of the parameters as follows:

```
sherpa> a1.nH=0.0336676
sherpa> b1.kT=20
sherpa> b1.norm=1e-02
sherpa> b2.kT=0.5
sherpa> b2.norm.min=1e-6
sherpa> b2.norm=1e-05
```

Note that the normalization of b2, "b2.norm", appears to have a better fit at values less than the default minimum. We set the b2.norm.min to a new, smaller value to give us the optimal fit. Now we have a reasonable start to the fit:

```
sherpa> LPLOT 2 FIT BACKFIT
```

The plot is visible here 📷.

In general, we may also want to change our fit <u>STATISTIC</u> or the <u>METHOD</u>. Now, run the fit:

```
sherpa> FIT
 LVMQT: V2.0
 LVMQT: initial statistic value = 2082.77
 LVMQT: final statistic value = 946.229 at iteration 5
          a1.nH  0.033839  10^22/cm^2
          b1.kT  20  keV
          b1.norm  0.00953351  L39/(D10)**2
          b2.kT  0.563334  keV
          b2.norm  8.06402e-06  L39/(D10)**2
```

Once the fit is complete, *Sherpa* will display the fit values to the screen. You may also display the overall status of the *Sherpa* session with the SHOW command:

```
sherpa> SHOW

Optimization Method: Levenberg-Marquardt
Statistic:           Chi-Squared Gehrels

-----------------
Input data files:
-----------------

Data 1: source.pi pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 10558
Exposure: 49429.23 sec
Count rate: 0.214 cts/sec
  Backscal: 1.872535e-05

  Background 1: back.pi pha.
  Total Size: 1024 bins (or pixels)
  Dimensions: 1
  Total counts (or values): 5622
  Exposure: 49429.23 sec
  Count rate: 0.114 cts/sec
  Backscal: 2.696451e-05

  The data are NOT background subtracted.

Current filters for dataset 1:
IGNORE source 1 energy : 0.3 , 10 :
Noticed filter size: 663 bins

Sum of data within filter: 9427

IGNORE back 1 energy : 0.3 , 10 :
Noticed filter size: 663 bins

Sum of data within filter: 5597

-----------------------------
Defined analysis model stacks:
-----------------------------

source 1 = (a1 * b1)
instrument source 1 = r1
bg 1 = (a1 * b2)
instrument back 1 = r2

-------------------------------------------
Defined source/background model components:
-------------------------------------------

xswabs[a1]  (XSPEC model name: absw)  (integrate: off)
    Param   Type       Value        Min         Max             Units
    -----   ----       -----        ---         ---             -----
 1    nH thawed 3.3839e-02      1e-07          10          10^22/cm^2

xsbbody[b1]  (XSPEC model name: bbody)  (integrate: on)
    Param   Type       Value        Min         Max             Units
    -----   ----       -----        ---         ---             -----
 1    kT thawed         20      1e-04         200                 keV
 2  norm thawed 9.5335e-03 2.6097e-05       0.261         L39/(D10)**2

xsbbody[b2]  (XSPEC model name: bbody)  (integrate: on)
```

```
     Param    Type       Value          Min          Max                 Units
     -----    ----       -----          ---          ---                 -----
 1      kT thawed       0.5633       1e-04          200                   keV
 2    norm thawed     8.064e-06      1e-06        0.261          L39/(D10)**2


----------------------------------
Defined instrument model components:
----------------------------------


rsp1d[r1]
    Param    Type   Value
    -----    ----   -----
 1     rmf string: "source.rmf" (N_E=1090,N_PHA=1024)
 2     arf string: "source.arf" (N_E=1090)

rsp1d[r2]
    Param    Type   Value
    -----    ----   -----
 1     rmf string: "back.rmf" (N_E=1090,N_PHA=1024)
 2     arf string: "back.arf" (N_E=1090)
```

One may visually examine the fit as well:

```
sherpa> LPLOT 2 FIT BACKFIT
```

The plot is visible here 📷.

Summary

This thread is complete, so we can exit the *Sherpa* session:

```
sherpa> EXIT
Goodbye.
```

History

14 Jan 2005 updated for CIAO 3.2: minor change in fit results
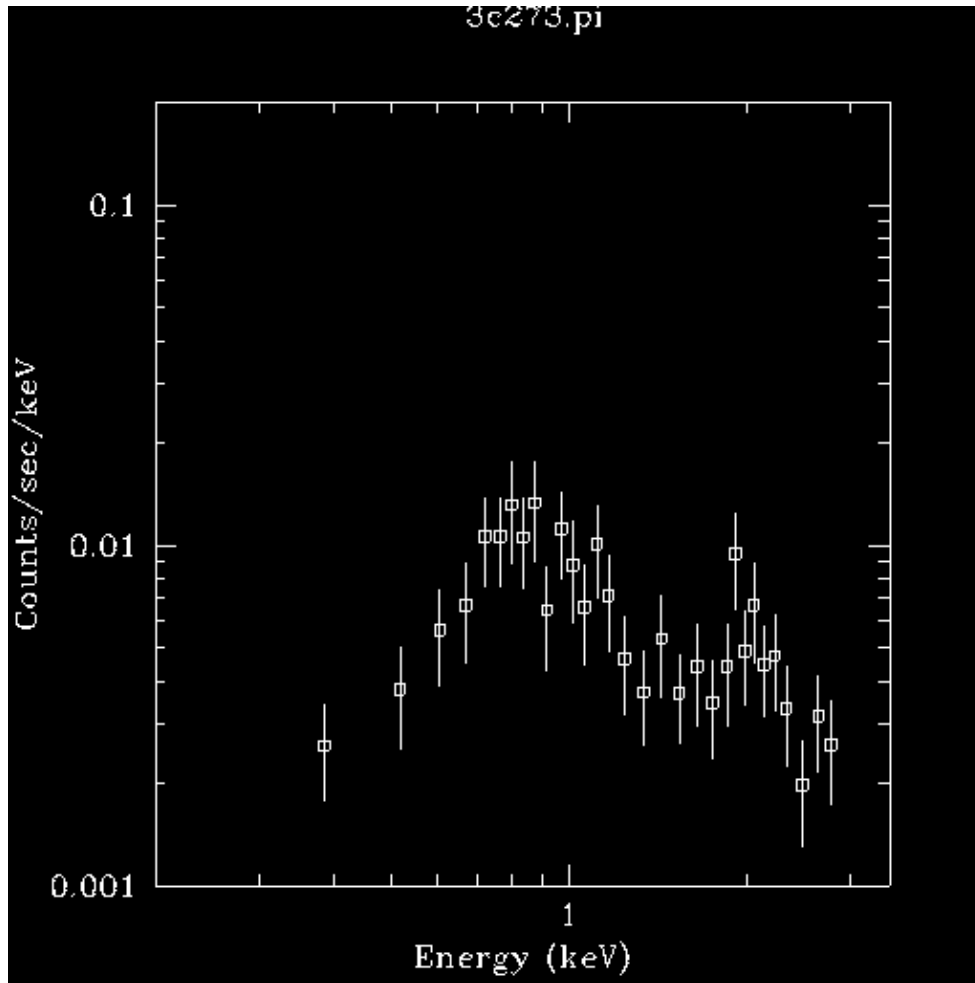
21 Dec 2005 reviewed for CIAO 3.3: no changes

01 Dec 2006 reviewed for CIAO 3.4: no changes
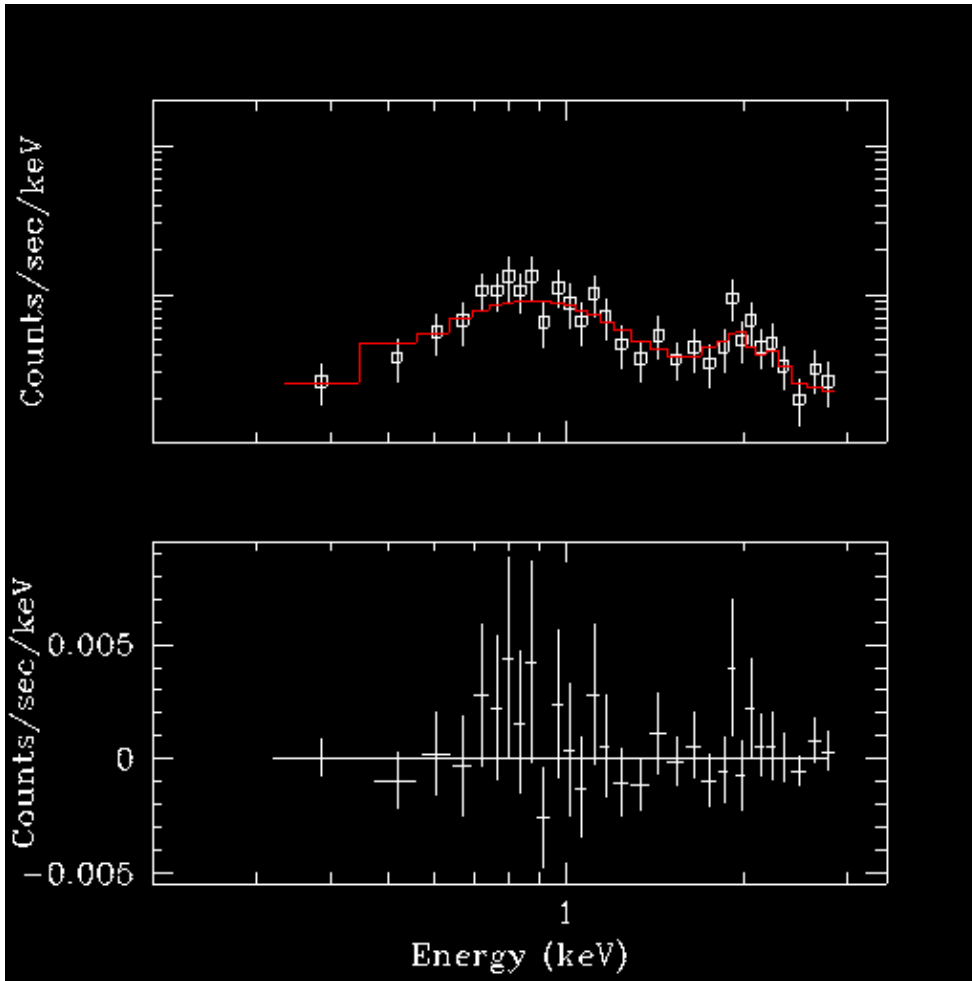
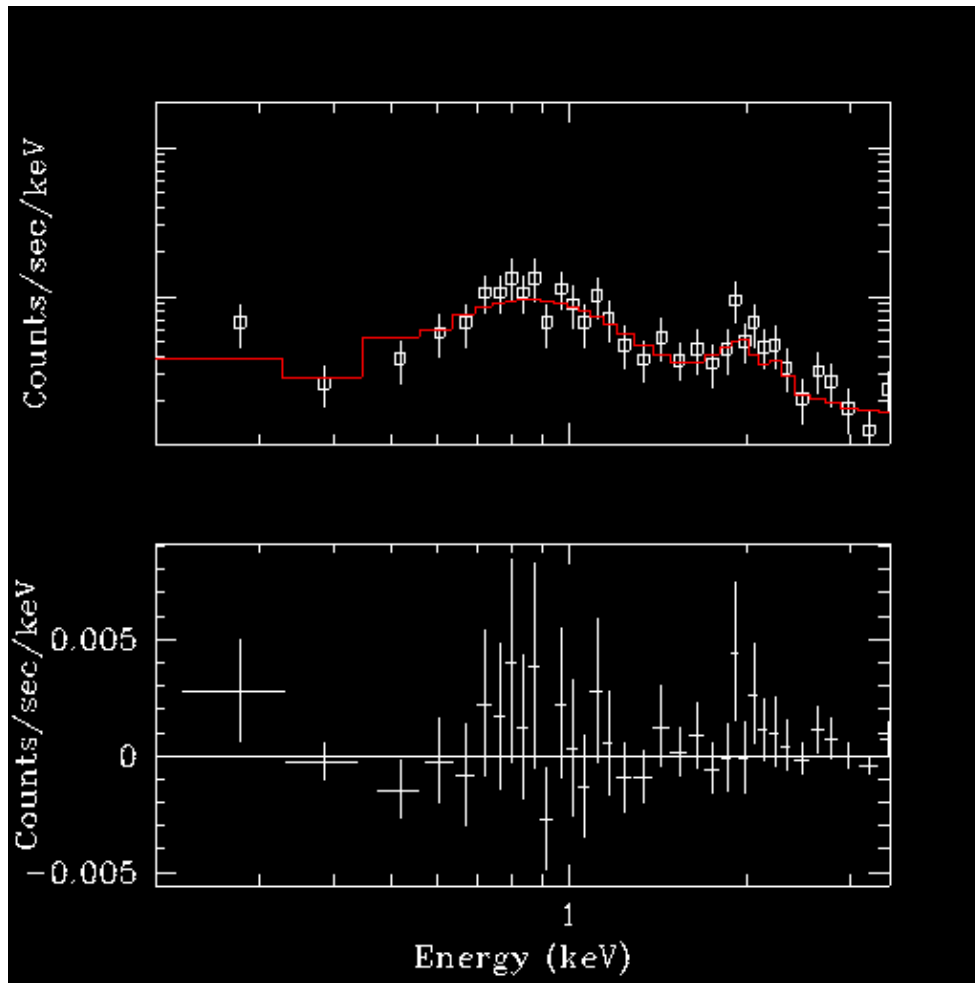URL: http://cxc.harvard.edu/sherpa/threads/sourceandbg/   Last modified: 1 Dec 2006

Image 1: The background–subtracted source spectrum
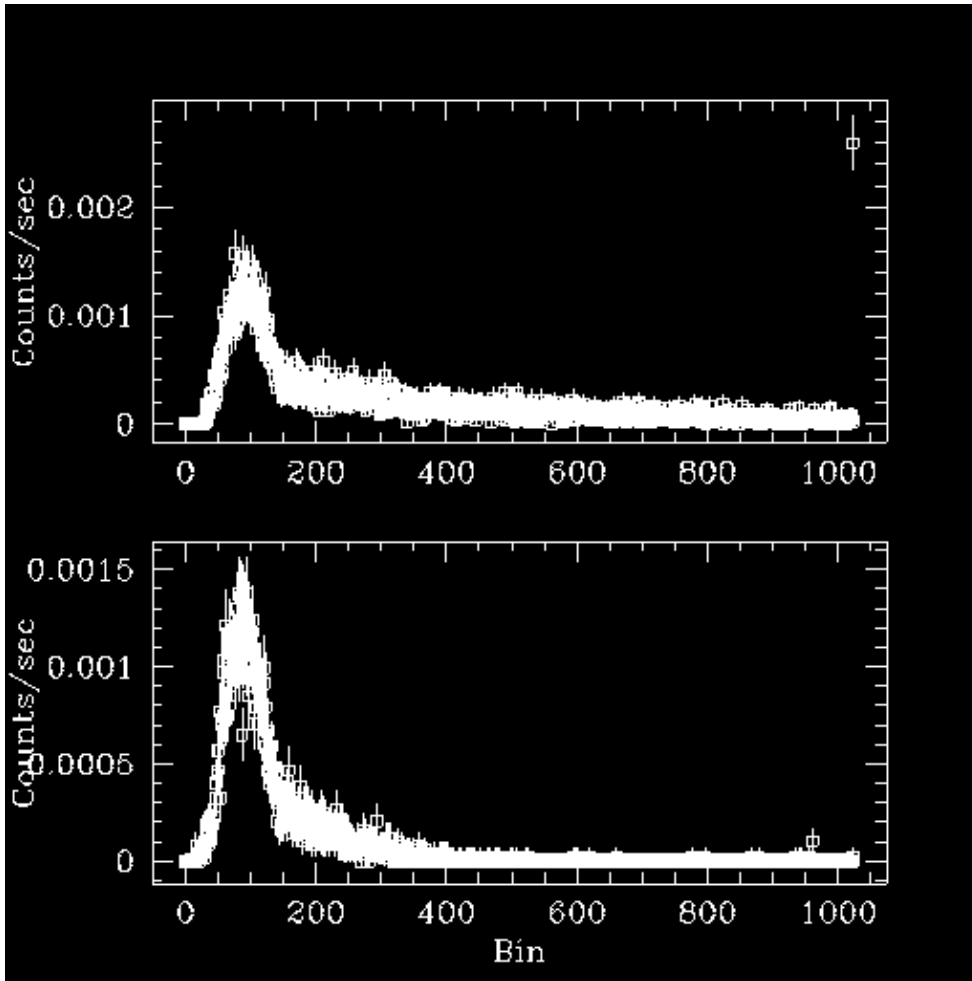
Image 2: Fit of the background–subtracted data

Image 2: Fit of the background–subtracted data

Image 3: Simultaneous fit of the source and background data (same responses)

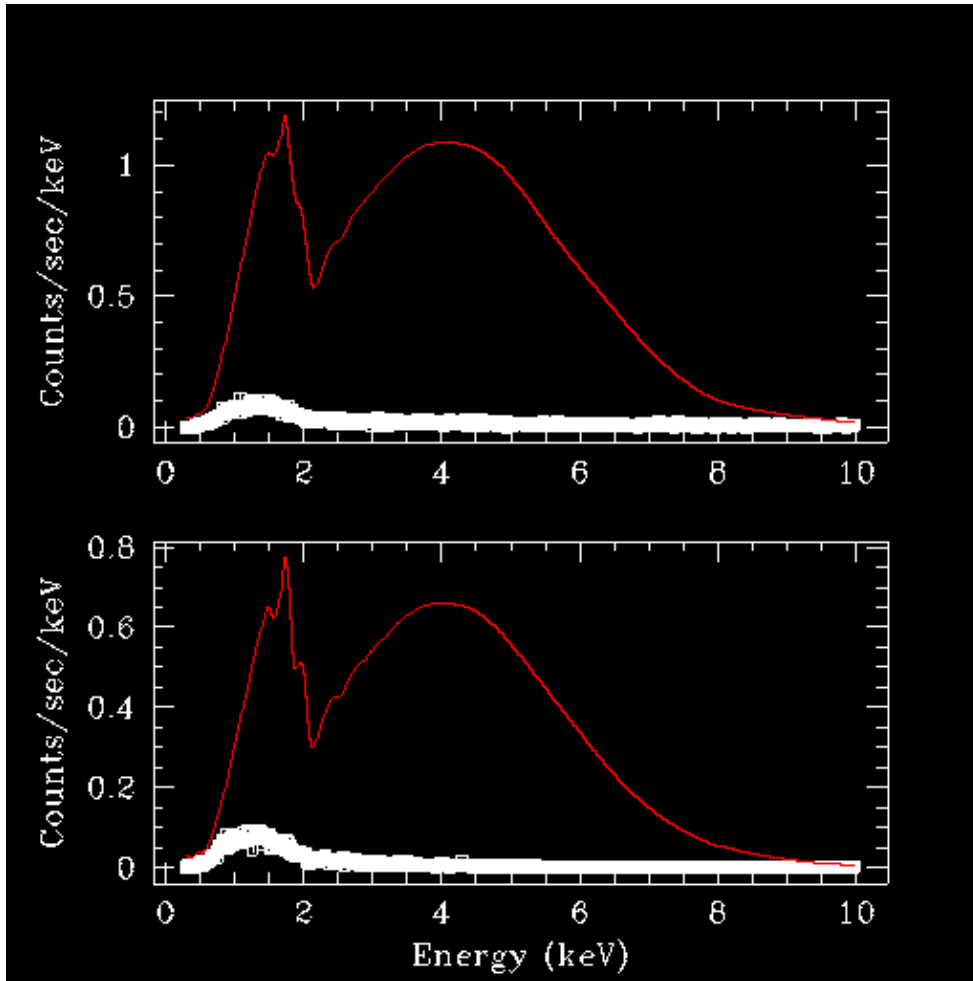# Image 4: Plots of the ungrouped source and background data

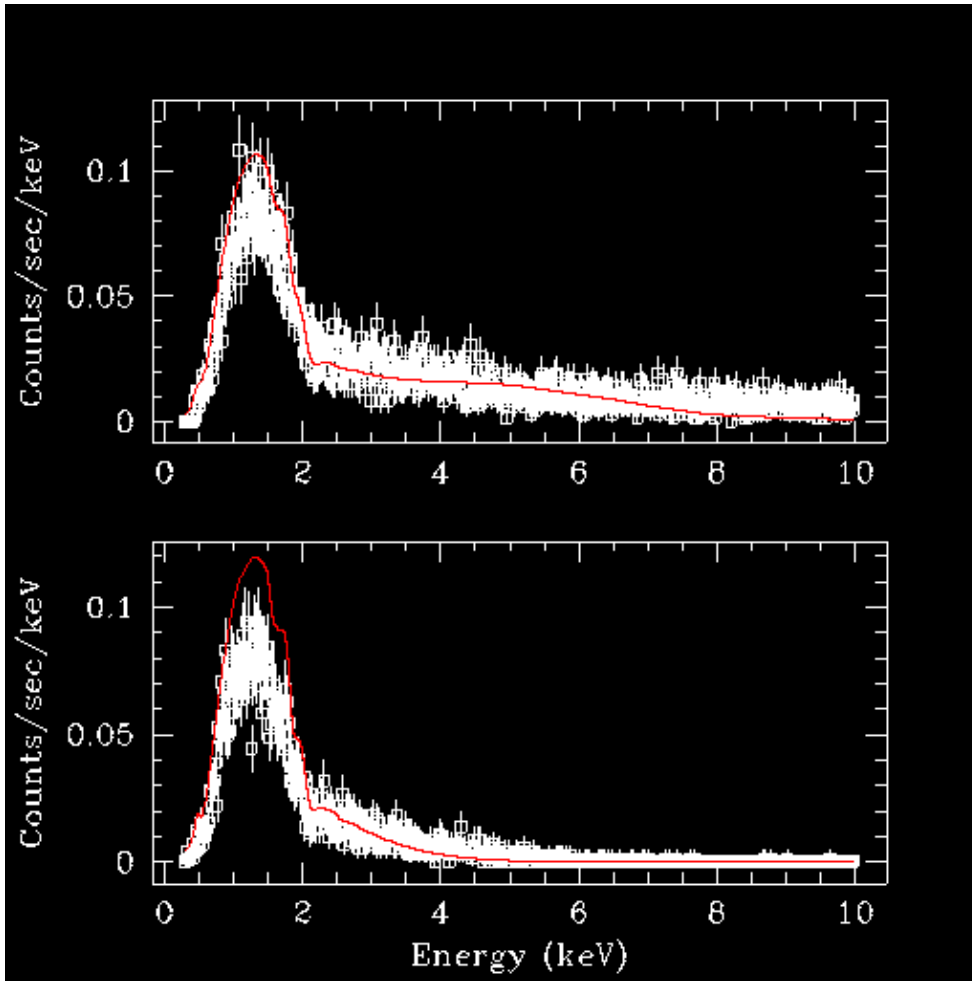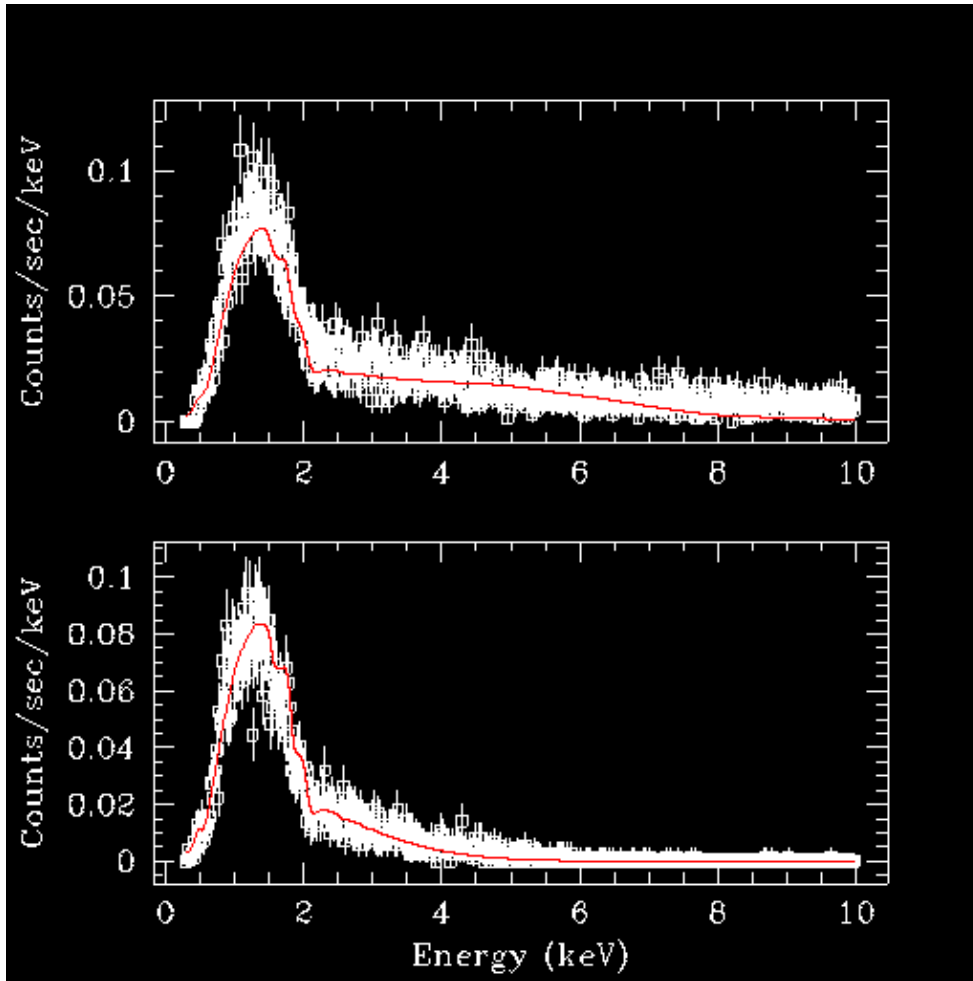Image 4: Plots of the ungrouped source and background data

Image 5: Plots of the ungrouped source and background data with default fits

Image 6: Plots of the ungrouped source and background data with custom fits

Image 6: Plots of the ungrouped source and background data with custom fits

Image 7: Simultaneous fit of the ungrouped source and background data (different responses)

20   Image 7: Simultaneous fit of the ungrouped source and background data (different responses)