SHERPA
The Modeling and Fitting Tool of the CIAO Software System

Sherpa Team:

Stephen Doe
Peter Freeman
Holly Jessop
Aneta Siemiginowska

Bill Joye (SAOFIT: Spring 1997)
Malin Ljungberg (ASCFIT 1997-99)
Mark Birkinshaw - OPTIM
This is NOT a Sherpa DEMO!

- Enhancements to Sherpa in CIAO 2.0
- Main Components of Sherpa
- Models and Model Language
- Statistics and Optimization
- Future Plans
SHERPA in CIAO 2.0

- 18 new commands including e.g. COVARIANCE, PROJECTION, SETBACK, WCS, LPLOT
- 1D modeling in wavelength, energy or bin space e.g. ANALYSIS
- Global filtering allows for filtering several data sets with one command (ALLSETS).
- GUIDE – Grating User Interactive Data Extension

```python
sherpa> import(`'guide'``)
```
- MDL (Model Descriptor List) FITS file created with WRITE command stores fit results.
- Sherpa preference file: `.sherparc` can be used to store defaults.
- Simultaneous fitting of data specified with DATA and BACK. Two backgrounds are allowed (important for grating analysis).
- Error Analysis includes now new algorithms e.g. COVARIANCE and PROJECTION. Visualization has been improved significantly.
- Updated Optimization methods are more robust and convergence has been improved.
- New Sherpa Models:
  - BPL/BPL1D: broken power law
  - CONST/CONST1D & CONST2D a positive constant,
  - DELTA/DELTA1D & DELTA2D: a delta function
  - EDGE: absorption edge as a function of energy or wavelength 1D
  - LINEBROAD: a line-broadening model 1D
  - NGAUSS/NGAUSS1D: a normalized Gaussian model 1D
Main SHERPA Components

- Data Input/Output.
- Visualization through ChIPS and ds9.
- Model library and model language.
- Statistics and Error Analysis.
- Optimization Methods.
Data Input/Output

- General use of data type and dimensionality.
- Supported types of files: ASCII, FITS binary tables and Images, PHA types I & II, IRAF IMH and QPOE files.
- Sherpa:
  - groups the data if appropriate;
  - treats integer, float or double precision data;
  - supports data of arbitrary dimensionality
- Data Model gives a main I/O interface.
- Filtering while reading the data.
- Input data on the command line in two ways.
sherpa> data "image.fits[150:300,160:310]"
sherpa> show
Current Data Files:
Total Size: 22801 bins (or pixels)
Dimensions: 2
Size: 151 x 151
Total counts (or values): 20711 cts

or

sherpa> mydata=readfile("image.fits[150:300,160:310]")
sherpa> print(mydata)
filename = image.fits
path = /data
filter = [150:300,160:310]
naxes = 2
transform = TAN
datatype = Real4
pixels = Float_Type[151,151]
crval = Double_Type[2]
crpix = Double_Type[2]
crdelt = Double_Type[2]
sherpa> print(mydata.crval[0])
278.386
sherpa> print(mydata.crval[1])
-10.5899
MODELS

• Three main type of models:
  
  Source
  Background
  Instrument

• Model library consists of several models (plus XSPEC v.11) which can be used to define a source or background model.

• There are three type of instrument models:
  
  RSP
  PsfFromTCD
  PsfFromFile

• Instrument models are convolved with Source and Background models before the model predicted data is compared with the observed data.

• Instrument and Background models are NOT required. Source models have to be defined for fitting.
RSP[rsp]

RMF file name:

ARF file name:

EEARF file name:

PSFfromTCD[psffromtcd]

<table>
<thead>
<tr>
<th>Param</th>
<th>Type</th>
<th>Value</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>numCuts frozen</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>convTyp frozen</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>xsize frozen</td>
<td>4</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>ysize frozen</td>
<td>4</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>nsigma frozen</td>
<td>2</td>
<td>0.0100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>funcTyp frozen</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>norm frozen</td>
<td>1</td>
<td>0</td>
<td>3.4028e+38</td>
<td></td>
</tr>
<tr>
<td>dim frozen</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

The Function Type is: Gaussian.

Dimension: 1

PSFfromFile[psffromFile]

<table>
<thead>
<tr>
<th>Param</th>
<th>Type</th>
<th>Value</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>numCuts frozen</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>convTyp frozen</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>file string:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xsize frozen</td>
<td>32</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>ysize frozen</td>
<td>32</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>xoff frozen</td>
<td>0</td>
<td>-512</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>yoff frozen</td>
<td>0</td>
<td>-512</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>xpos thawed</td>
<td>512</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>ypos thawed</td>
<td>512</td>
<td>1</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>norm frozen</td>
<td>1</td>
<td>0</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

Model Language

- All predefined in model library models can be used in model expression to build a **source or background model**.

- Each library model can be given a **unique name** within Sherpa session.

  ```
  sherpa> gauss1d[g1]
  sherpa> source = ATTEN[att1]*BPL[b1]
  att1.hcol parameter value [1e+20]
  att1.heiRatio parameter value [0.1]
  att1.heiiRatio parameter value [0.01]
  b1.gamma1 parameter value [0]
  b1.gamma2 parameter value [0]
  b1.eb parameter value [100]
  b1.ref parameter value [1]
  b1.ampl parameter value [1]
  ```

- Model Parameters can be **linked** to other model parameters, arithmetic expression or other models.

  ```
  sherpa> source = POLY[con]+gauss1d[g1]+gauss1d[g2]
  sherpa> g1.ampl => 0.4*g2.ampl
  ```

  or

  ```
  sherpa> func = const1d[red]
  sherpa> g1.pos => 0.568*func
  ```
• An argument of a model (e.g. energy) is defined as an expression in **Nested Models**.

```plaintext
Parameter Expression:
sherpa> Temperature = POLY
sherpa> BB.kT => Temperature
sherpa> show source
BB
bbbody[BB] (integrate: on)
<table>
<thead>
<tr>
<th>Param</th>
<th>Type</th>
<th>Value</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>kT</td>
<td>varying</td>
<td></td>
<td>expression: Temperature</td>
</tr>
<tr>
<td>2</td>
<td>ampl</td>
<td>thawed</td>
<td>0.3</td>
<td>1e-20</td>
</tr>
</tbody>
</table>

Argument Expression:
sherpa> xenergy = SHLOG[mod]
sherpa> source = BB{xenergy}
sherpa> show source
BB{ xenergy }
bbbody[BB] (integrate: on)
<table>
<thead>
<tr>
<th>Param</th>
<th>Type</th>
<th>Value</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>kT</td>
<td>thawed</td>
<td>0.3</td>
<td>0.1000</td>
</tr>
<tr>
<td>2</td>
<td>ampl</td>
<td>thawed</td>
<td>0.001</td>
<td>1e-20</td>
</tr>
</tbody>
</table>

shloge[mod] (integrate: off)
<table>
<thead>
<tr>
<th>Param</th>
<th>Type</th>
<th>Value</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>offset</td>
<td>frozen</td>
<td>0-3.4028e+38</td>
<td>3.4028e+38</td>
</tr>
<tr>
<td>2</td>
<td>coeff</td>
<td>frozen</td>
<td>1-3.4028e+38</td>
<td>3.4028e+38</td>
</tr>
<tr>
<td>3</td>
<td>ampl</td>
<td>frozen</td>
<td>1</td>
<td>0 3.4028e+38</td>
</tr>
</tbody>
</table>
```
• For **Joint-Mode** analysis one can apply models on each axis:

sherpa> DATA image.fits FITSIMAGE
sherpa> LORENTZ[ SpatialAxis0](98:5:200, 70:50:90, 1:1:200)
sherpa> POWLAW1D[ SpecAxis1]
sherpa> SRC = SpatialAxis0{x1}*SpecAxis1{x2}
sherpa> show source
(SpatialAxis0{ 0 } * SpecAxis1{ 1 })
lorentz1d[ SpatialAxis0] (integrate: on)

<table>
<thead>
<tr>
<th>Param</th>
<th>Type</th>
<th>Value</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>fwhm</td>
<td>thawed</td>
<td>98</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>pos</td>
<td>thawed</td>
<td>70</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>ampl</td>
<td>thawed</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

powlaw1d[ SpecAxis1] (integrate: on)

<table>
<thead>
<tr>
<th>Param</th>
<th>Type</th>
<th>Value</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>gamma</td>
<td>thawed</td>
<td>1.5</td>
<td>-10</td>
</tr>
<tr>
<td>2</td>
<td>ref</td>
<td>frozen</td>
<td>1 - 3.4028e+38</td>
<td>3.4028e+38</td>
</tr>
<tr>
<td>3</td>
<td>ampl</td>
<td>thawed</td>
<td>1</td>
<td>1e-20</td>
</tr>
</tbody>
</table>