Introduction to SHERPA

The Modeling and Fitting Tool
of the CIAO Software System

Aneta Siemiginowska

What can you do in Sherpa?

- Standard PHA based analysis.
- Model data in many spectral bands simultaneously, e.g optical/X-rays.
- Access ATOMDB and GUIDE/ISIS for grating data analysis.
- Fit radial profiles.
- Simulate 1D data.
- Model 2D image data, e.g. fit surface brightness of the extended source.
- Get normalization of your PSF, while fitting the data with 1D/2D PSF.
- Use the PSF as a convolution kernel in the 2D image analysis (FFT or sliding cell).
- Convolution using the TCD library kernel.
- Use of exposure maps in the image analysis.
- Joint-mode data: spatial-spectral, spatial-timing
- Use scripts based on Sherpa only commands.
- Use S-lang on command line and in S-lang based scripts.
- Use your own models with User Models and S-lang user models.

Standard PHA based analysis:

- Source data:
  - can be modeled in energy/wavelength space.
  - multiple data sets can be modeled with the same or different models in one Sherpa session.
  - data can be filtered on the command line, or from filter file.
- Instrument responses (RMF/ARF):
  - are entered independently from the source data.
  - one set of instrument responses can be read once and applied to multiple data sets.
  - several instrument responses and use in analysis of one source model or multiple data sets.
  - Future: multiple response files can be used in one source model expression.
- Background files:
  - are entered independently from the source data.
  - multiple background files can be used for one data set, e.g. grating analysis
  - the same background can be applied to multiple data sets.
  - background can be modeled independently of the source data, and have its separate instrument responses.
  - background can be modeled simultaneously with the source data.
  - background can be subtracted from the source data (subtract/unsubtract).

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
- I/O interface through Data Model and Varmm
- 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:
Data 1: image.fits[150:300,160:310] fits.
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 different types 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]
    Param   Type      Value          Min          Max
    -----   ----      -----          ---          ---
 1numCuts frozen          1            1            1
 2convTyp frozen          1            1            2
 3  xsize frozen          4            1         1024
 4  ysize frozen          4            1         1024
 5 nsigma frozen          2       0.0100          100
 6funcTyp frozen          1            0            7
 7   norm frozen          1            0 3.4028e+38
 8    dim frozen          1            0            7
 The Function Type is:  Gaussian.
 Dimension: 1
PSFfromFile[psffromFile]
    Param   Type      Value          Min          Max
    -----   ----      -----          ---          ---
 1numCuts frozen          1            1            1
 2convTyp frozen          1            1            2
 3   file string:
 4  xsize frozen         32            1         1024
 5  ysize frozen         32            1         1024
 6   xoff frozen          0         -512          512
 7   yoff frozen          0         -512          512
 8   xpos thawed        512            1         1024
 9   ypos thawed        512            1         1024
10   norm frozen          1            0         1000
```

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.

```
Parameter Expression:
sherpa> Temperature = POLY
sherpa> BB.kT => Temperature
sherpa> show source
BB
bbody[BB]  (integrate: on)
    Param   Type      Value        Min        Max
    -----   ----      -----        ---        ---
 1     kT   link    varying     expression: Temperature
 2   ampl thawed        0.3      1e-20 3.4028e+38

Argument Expression:
sherpa> xenergy = SHLOG[mod]
sherpa> source = BB{xenergy}
sherpa> show source
BB{ xenergy }
bbody[BB]  (integrate: on)
    Param   Type      Value        Min        Max
    -----   ----      -----        ---        ---
 1     kT thawed        0.3     0.1000       1000
 2   ampl thawed      0.001      1e-20 3.4028e+38
shloge[mod]  (integrate: off)
    Param   Type      Value        Min        Max
    -----   ----      -----        ---        ---
 1 offset frozen          0-3.4028e+38 3.4028e+38
 2  coeff frozen          1-3.4028e+38 3.4028e+38
 3   ampl frozen          1          0 3.4028e+38
```

- 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)
    Param   Type      Value         Min         Max
    -----   ----      -----         ---         ---
 1   fwhm thawed         98           5         200
 2    pos thawed         70          50          90
 3   ampl thawed          1           1         200
powlaw1d[SpecAxis1]  (integrate: on)
    Param   Type      Value         Min         Max
    -----   ----      -----         ---         ---
 1  gamma thawed        1.5         -10          10
 2    ref frozen          1-3.4028e+38 3.4028e+38
 3   ampl thawed          1       1e-20 3.4028e+38
```

Fit Statistics in Sherpa:

A key feature of *Sherpa* is its large array of statistics appropriate for analyzing Poisson-distributed (*i.e.* counts) data.

- Statistics based on χ^2:
  - `CHI GEHRELS`
  - `CHI DVAR`
  - `CHI MVAR`
  - `CHI PARENT`
  - `CHI PRIMINI`
- Statistics based on the Poisson likelihood $\mathcal{L}$:
  - `CASH`
  - `BAYES`

If the data are not Poisson-distributed (*e.g.* fluxes), then alternatives include:

- least-squares fitting: setting all variances to one; or
- providing errors in an input file.

χ^2-Based Statistics

The χ^2 statistic is

$$\chi^2 \equiv \sum_i \frac{(D_i - M_i)^2}{\sigma_i^2},$$

where

- D_i represents the observed datum in bin i;
- M_i represents the predicted model counts in bin i; and
- σ_i^2 represents the variance of the sampling distribution for D_i.

χ^2 Statistic

σ_i^2

GEHRELS: $[1 + \sqrt{D_i + 0.75}]^2$

DVAR D_i

MVAR M_i

PARENT $\frac{\sum_{i=1}^{N} D_i}{N}$

PRIMINI M_i from previous best-fit

Likelihood-Based Statistics

The `CASH` statistic is

$$C \equiv 2\sum_i [M_i - D_i \log M_i] \propto -2 \log \mathcal{L}\,,$$

where

- D_i represents the observed datum in bin i;
- M_i represents the predicted model counts in bin i; and
- $\mathcal{L} = \Pi_i \frac{M_i^{D_i}}{D_i!} \exp(-M_i)$.