Sherpa

Aneta Siemiginowska & Omar Laurino

Chandra X-ray Center

http://cxc.harvard.edu/sherpa

PyGamma19 - March 21, 2019

Sherpa Team

Core Team (Chandra X-ray Center):

Omar Laurino
Doug Burke
Dan Nguyen
Warren McLaughlin
Aneta Siemiginowska

Code contributions:

Tom Aldcroft
Jamie Budynkiewicz
Christoph Deil
Rafael Martinez Galarza
Brigitta Sipocz

Sherpa - Summary

- Modeling and fitting application for Python.
- User Interface and high level functions written in Python.
- Modeling 1D/2D (N-D) data: arrays, spectra, images.
- Powerful language for building complex expressions.
- Provides a variety of statistics and optimization methods (including Bayesian analysis) .
- Support for wcs, responses, psf, convolution.
- Extensible to include user models, statistics and optimization methods.
- Included in several software packages.
- Source code on GitHub https://github.com/sherpa/sherpa
- Open development with continuous integration via Travis

Sherpa 20 years Milestones

December 1999

Sherpa release in CIAO 1.1

December 2000

Sherpa release with CIAO 2.0

- internal parser
- S-lang interpreted language

C++/ Fortran

December 2007

Sherpa Python - Beta release in CIAO 4.0

December 2008

Sherpa Python release in CIAO 4.1

April 2015

Sherpa development moved to GitHub

Standalone (non-CIAO) Sherpa release

Python API

C++/Fortran

Sherpa

Astronomical Data Analysis Software and Systems V
ASP Conference Series, Vol. 101, 1996
George H. Jacoby and Jeannette Barnes, eds.

1996

The ASC Fitting Environment

S. Doe, M. Conroy, J. McDowell

Smithsonian Astrophysical Observatory, MS 81, 60 Garden St., Cambridge, MA 02138 USA

Abstract. Fitting models to data will be a crucial part of both the calibration of the AXAF instruments and the scientific analysis of AXAF

Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

1997

Fitting and Modeling in the ASC Data Analysis Environment

S. Doe, A. Siemiginowska, W. Joye, and J. McDowell

Smithsonian Astrophysical Observatory, 60 Garden Street, MS 81, Cambridge, MA 02138

Astronomical Data Analysis Software and Systems VII
ASP Conference Series, Vol. 145, 1998
R. Albrecht, R. N. Hook and H. A. Bushouse, eds.

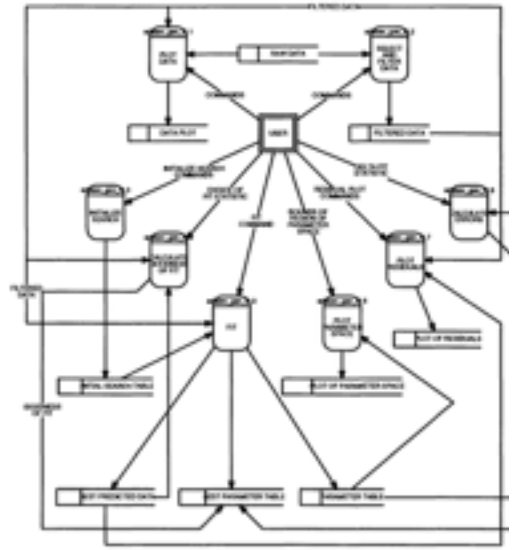
1998

Fitting and Modeling of AXAF Data with the ASC Fitting Application

S. Doe, M. Ljungberg, A. Siemiginowska and W. Joye

AXAF Science Center, MS 81, Smithsonian Astrophysical Observatory, 60 Garden Street, Cambridge, MA 02138 USA

ASC Fitting Environment



1. The ASC Fitting Environment.

Doe, Ljungberg, Siemiginowska and Joye

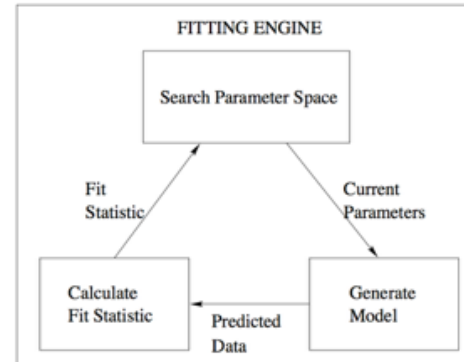


Figure 1. The main fitting process.

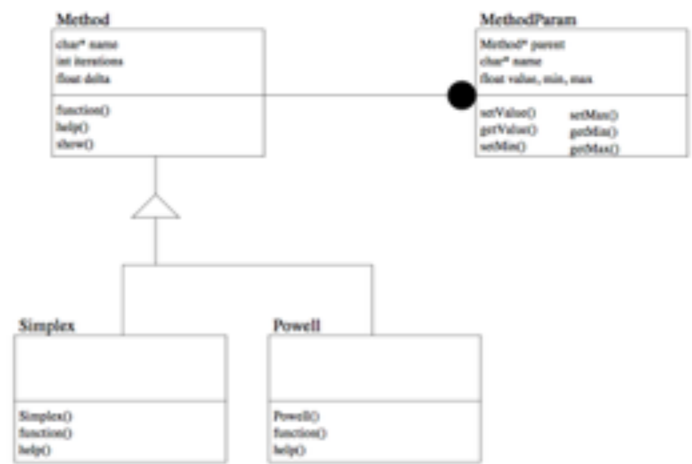


Figure 2. The Method Class.

Astronomical Data Analysis Software and Systems XVI
ASP Conference Series, Vol. 376, 2007
R. A. Shaw, F. Hill and D. J. Bell, eds.

2007

Developing Sherpa with Python

S. Doe, D. Nguyen, C. Stawarz, B. Refsdal, A. Siemiginowska, D. Burke,
I. Evans, J. Evans, J. McDowell

Smithsonian Astrophysical Observatory, Cambridge, MA, USA

J. Houck, M. Nowak

MIT Kavli Institute for Astrophysics and Space Research, Cambridge,

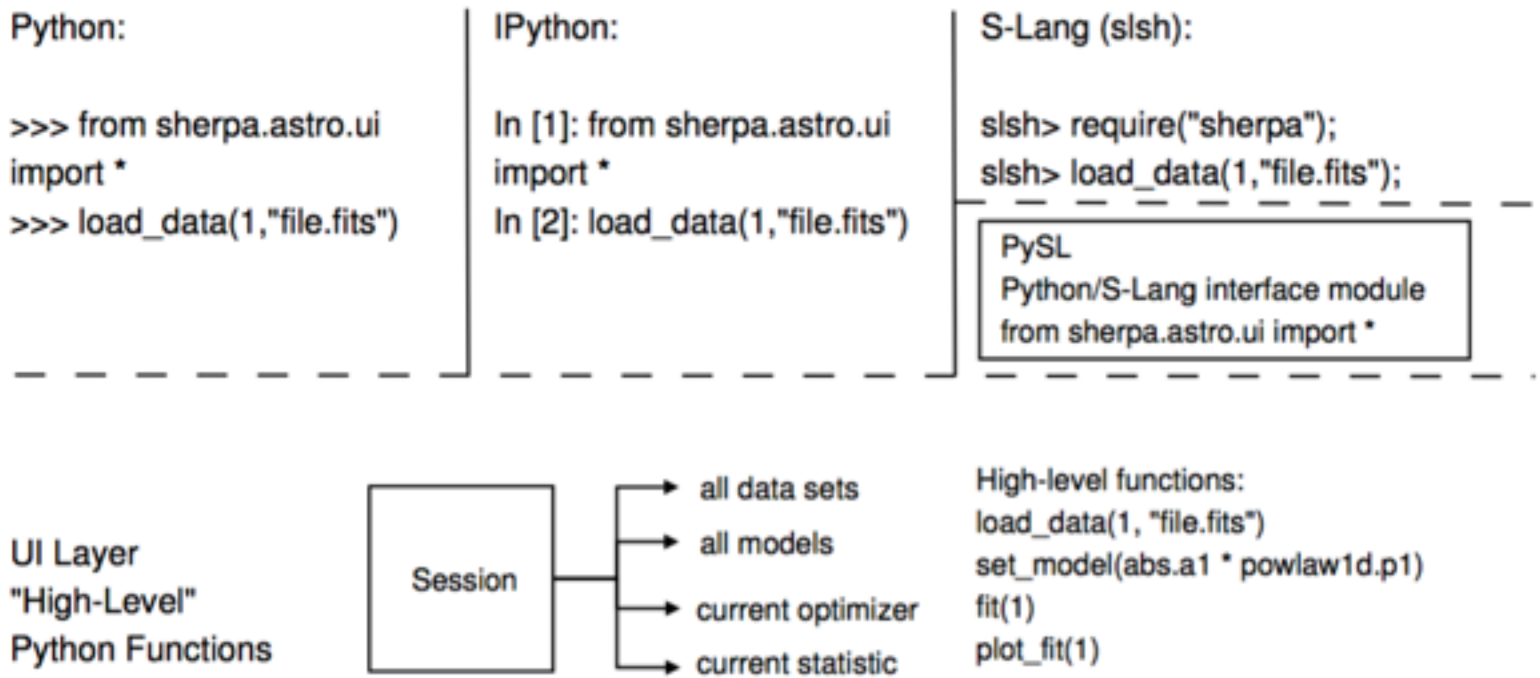


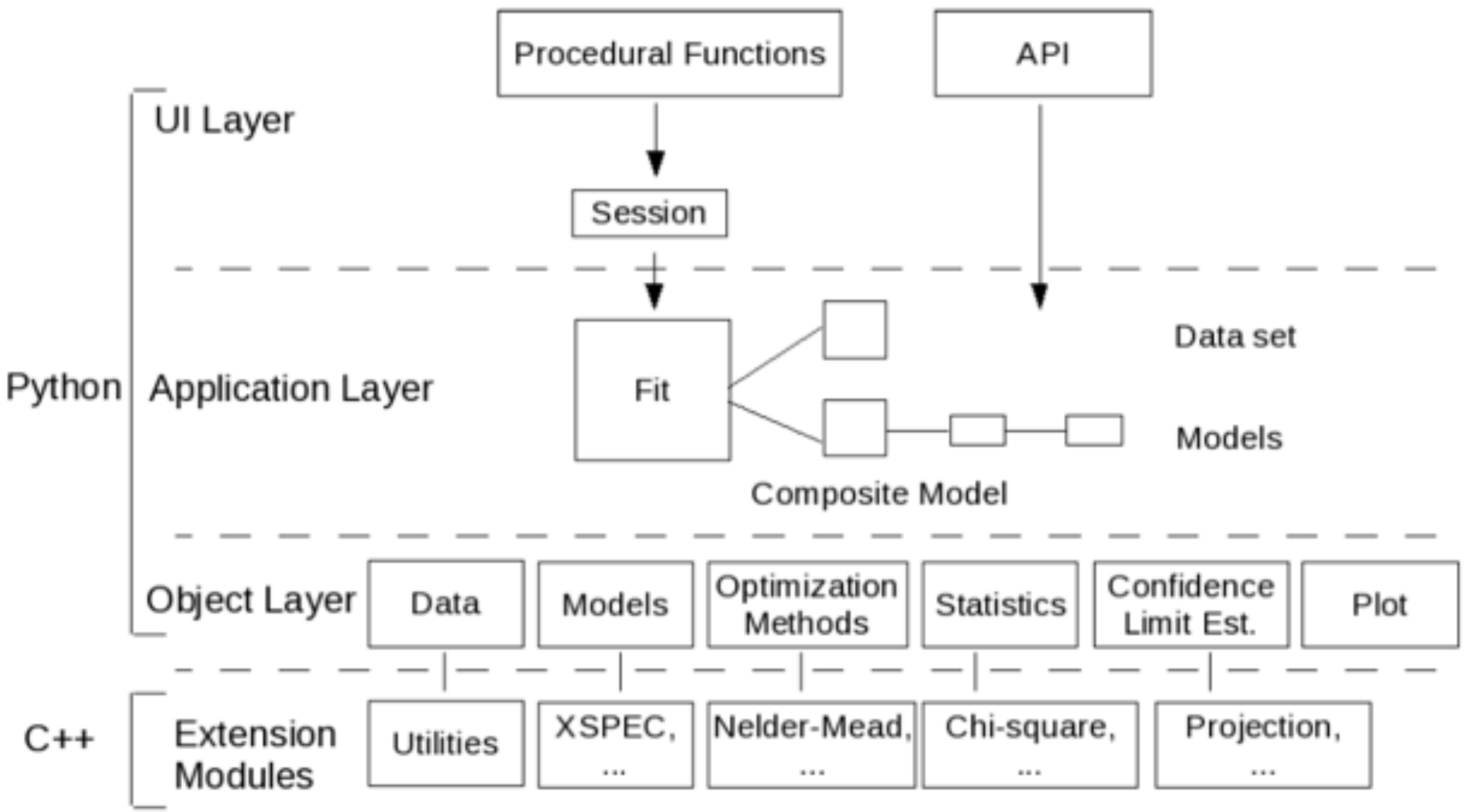
Figure 2. *Sherpa* modules in a variety of environments.

Proceedings of the 8th Python in Science Conference (SciPy 2009)

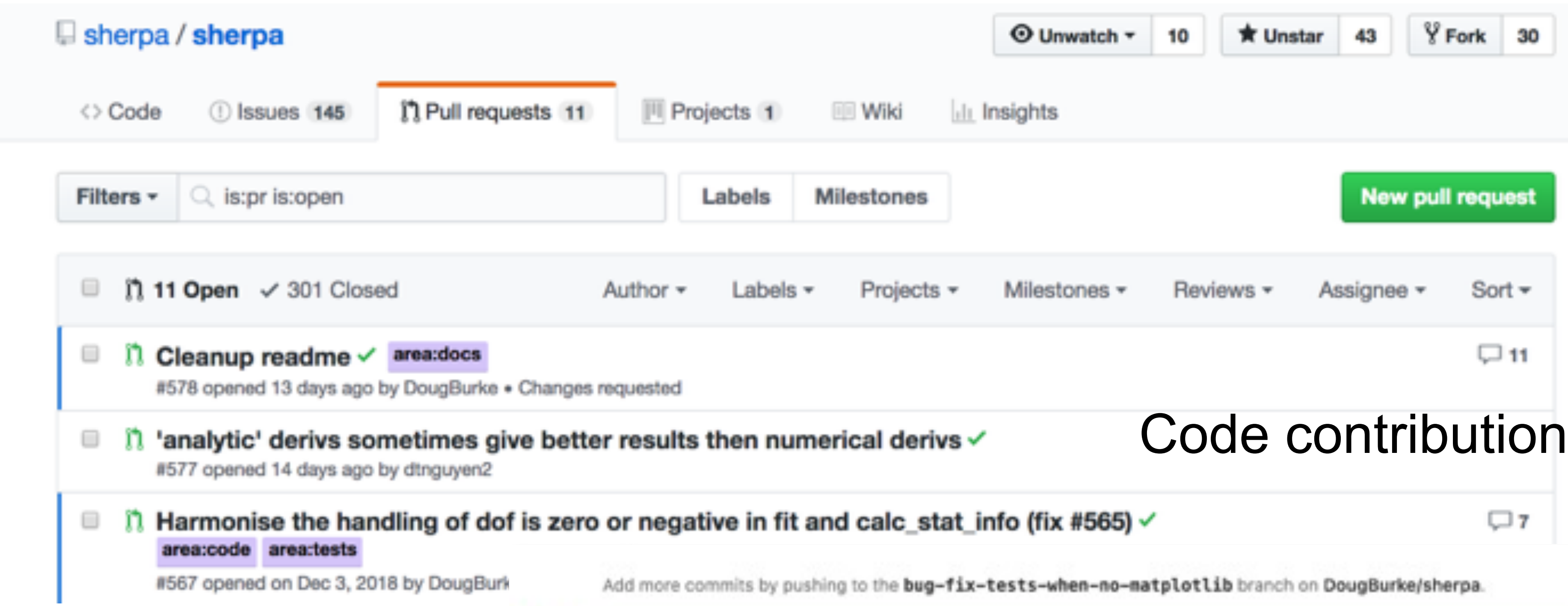
Sherpa: 1D/2D modeling and fitting in Python

2009

Brian L. Refsdal (brefsdal@head.cfa.harvard.edu) – *Harvard-Smithsonian Center for Astrophysics*, USA
Stephen M. Doe (sdoe@head.cfa.harvard.edu) – *Harvard-Smithsonian Center for Astrophysics*, USA
Dan T. Nguyen (dtn@head.cfa.harvard.edu) – *Harvard-Smithsonian Center for Astrophysics*, USA
Aneta L. Siemiginowska (aneta@head.cfa.harvard.edu) – *Harvard-Smithsonian Center for Astrophysics*, USA



Open Development on GitHub



Code contribution

Travis
- continuous integration

Sherpa

Data Input/Output
Astropy.io
PyCrates

Model Library
Sherpa, XSPEC models, user models, templates

Fit Statistics: Poisson and Gaussian likelihood

Fit Methods:
minimization and sampling

Visualization:
ChIPS, ds9, matplotlib

Final Evaluation & Conclusions

Data in Sherpa

- X-ray Spectra
  typically PHA files with the RMF/ARF calibration files
- X-ray Images
  FITS images, exposure maps, PSF files
- Lightcurves
  FITS tables, ASCII files
- Derived functional description of the source:
    - Radial profile
    - Temperatures of stars
    - Source fluxes
- Concepts of Source and Background data
- Any data array that needs to be fit with a model

Data in Sherpa

- Load functions to input data:

  data: load_data, load_pha, load_arrays, load_ascii
  calibration: load_arf, load_rmf load_multi_arfs, load_multi_rmfs
  background: load_bkg, load_bkg_arf , load_bkg_rmf
  2D image: load_image, load_psf
  General type: load_table, load_table_model, load_user_model

- Multiple Datasets - data id

  Default data id =1
  load_data(2, “data2.dat”, ncols=3)

Help file:
load_data([id=1], filename, [options])
load_image([id=1], filename|IMAGECrate,[coord="logical"])

Examples:
load_data("src", "data.txt", ncols=3)

load_data("rprofile_mid.fits[cols RMID,SUR_BRI,SUR_BRI_ERR]")
load_data(“image.fits”)
load_image(“image.fits”, coord=“world”))

- Filtering the data
  load_data expressions
  notice/ignore commands in Sherpa

Examples:
notice(0.3,8)
notice2d("circle(275,275,50)")

Models in Sherpa

- Parameterized models: $f(x_i, p_k)$

  absorption - N_H

  photon index of a power law function - Γ

  blackbody temperature kT

- Library of models

sherpa> list_models()

['absorptionedge',
'absorptiongaussian',
'atten',
'bbody',
'bbodyfreq',
'beta1d',
'beta2d',
'blackbody',
....

- User Models can be added
- Model language to build compound model expressions

Building Models: Expressions

- Standard operations: + - * :
- Linking parameters: link()
- Convolution:
  - responses, arf & rmf files via standard I/O
  - PSF - an image file or a Sherpa model
  - load_conv() - a generic kernel from a file or defined by a Sherpa model

Sherpa

Building Models: Examples

- Building composite models:
  - models in the library: e.g. *powlaw1d, atten*
  - give a name for a model component in the expression:

```
set_source(1,'atten.abs1*atten.abs2*powlaw1d.p1')
set_source(2,'abs1*abs2*powlaw1d.p2')
```

- Building a model expression with convolved and unconvolved components:

```
set_full_model(1,'psf(gauss2d.g2)+const2d.c1')
```

Building Models: Examples

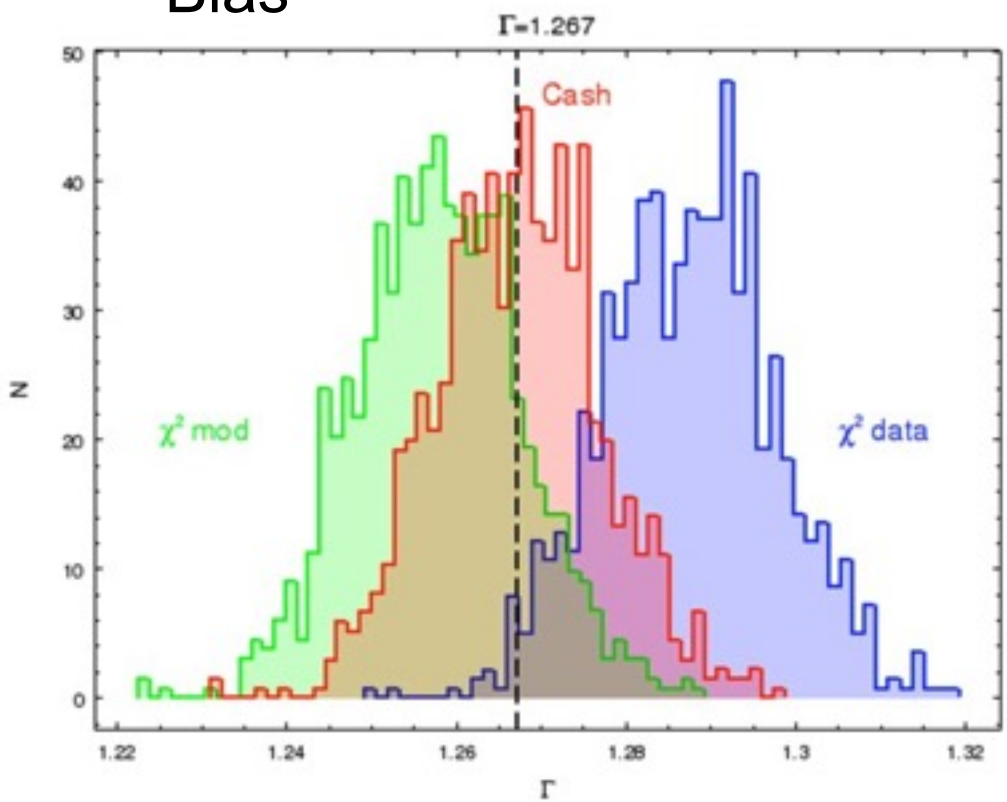
- Source and Background models:

```
set_source(2,'xsphabs.abs1*(powlaw1d.p1+gauss1d.g1)')
set_bkg_model(2,'const1d.mybkg')
```

Fit Statistics in Sherpa

```
In [19]: list_stats()
Out[19]:
['cash',
 'chi2',
 'chi2constvar',
 'chi2datavar',
 'chi2gehrels',
 'chi2modvar',
 'chi2xspecvar',
 'cstat',
 'leastsq',
 'userstat',
 'wstat']
In [20]: set_stat('cash')
```

Bias



"Handbook of X-ray Astronomy "
(2011), Arnaud, Smith, Siemiginowska

Optimization Methods in Sherpa

- "Single - shot" routines: Simplex and Levenberg-Marquardt

  start from a set of parameters, and then improve in a continuous fashion:
  - Very Quick
  - Depend critically on the initial parameter values
  - Investigate a local behaviour of the statistics near the initial parameters, and then make another guess at the best direction and distance to move to find a better minimum.
  - Continue until all directions result in increase of the statistics or a number of steps has been reached

- "Scatter-shot" routines: moncar (differential evolution)

  search over the entire permitted parameter space for a better minima than near the starting initial set of parameters.

- Bayesian sampling methods: Markov-Chain Monte Carlo

Sherpa

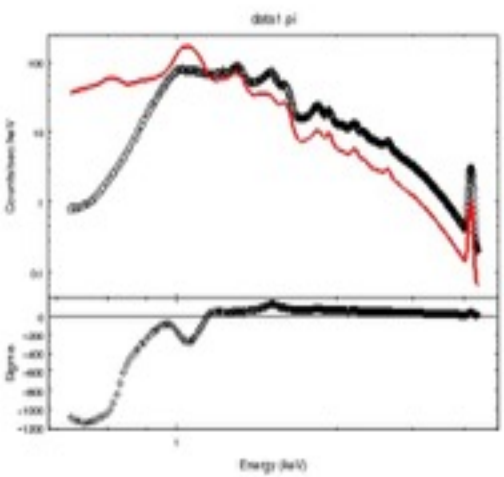
Optimization Methods: Comparison

Example: Spectral Fit with 3 methods

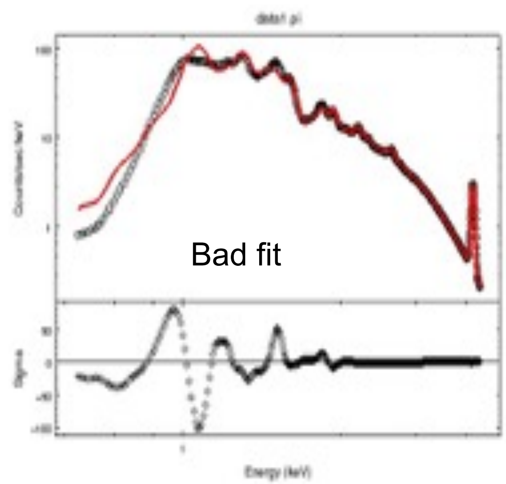
Data: high S/N simulated ACIS-S spectrum of the two temperature plasma
Model: photoelectric absorption plus two MEKAL components (correlated!)

Start fit from the same initial parameters
Figures and Table compares the efficiency
and final results

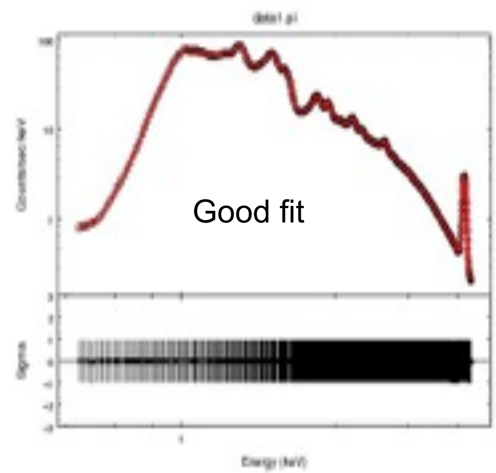
Method	Number of Iterations	Final Statistics
Levmar	31	1.55e5
Neldermead	1494	0.0542
Moncar	13045	0.0542



Data and Model with initial parameters



Levmar fit



Nelder-Mead and Moncar fit

Optimization Methods: Probing Parameter Space

Temperature

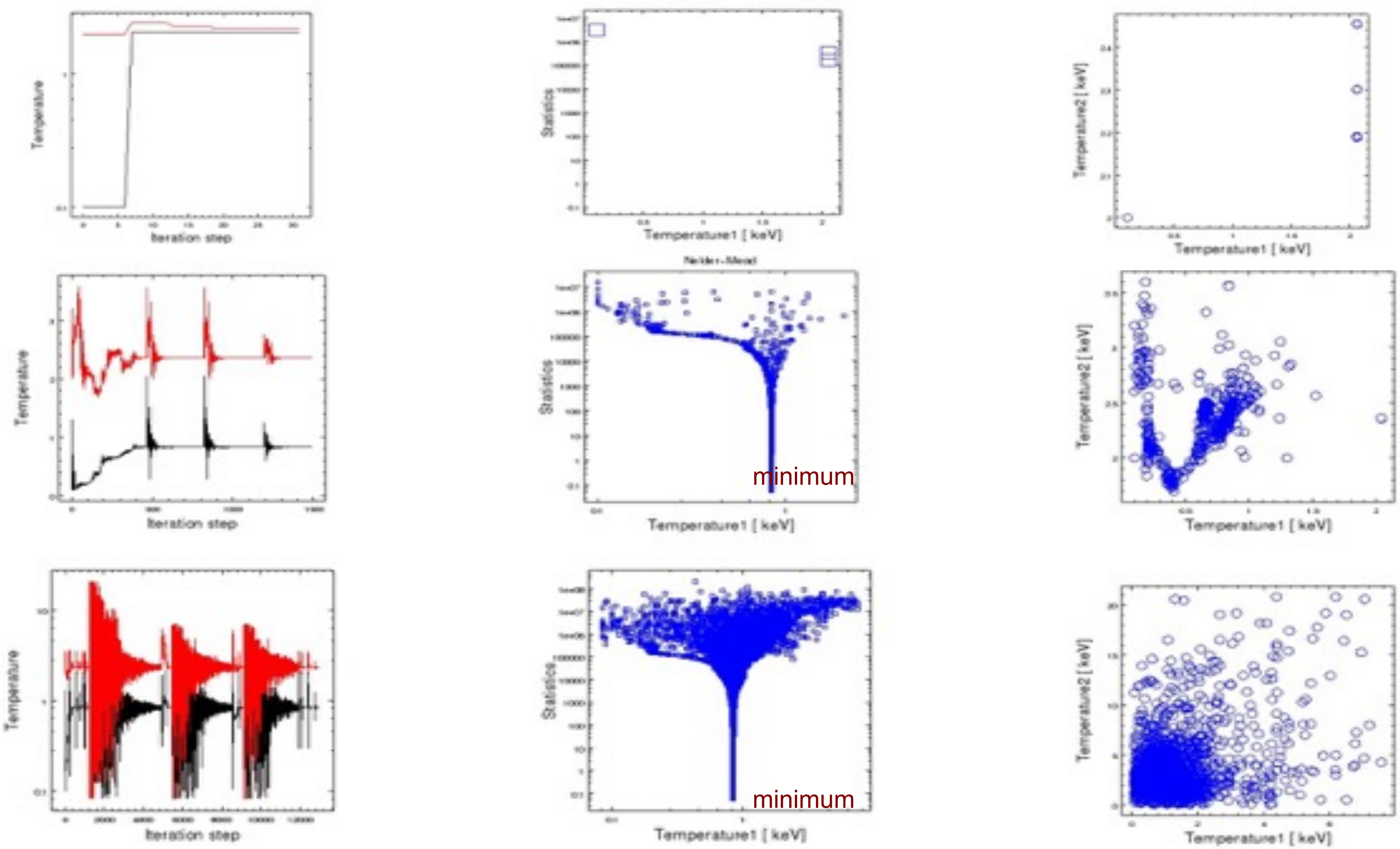
Statistics vs. Temperature

2D slice of Parameter Space probed by each method

levmar

simplex

moncar



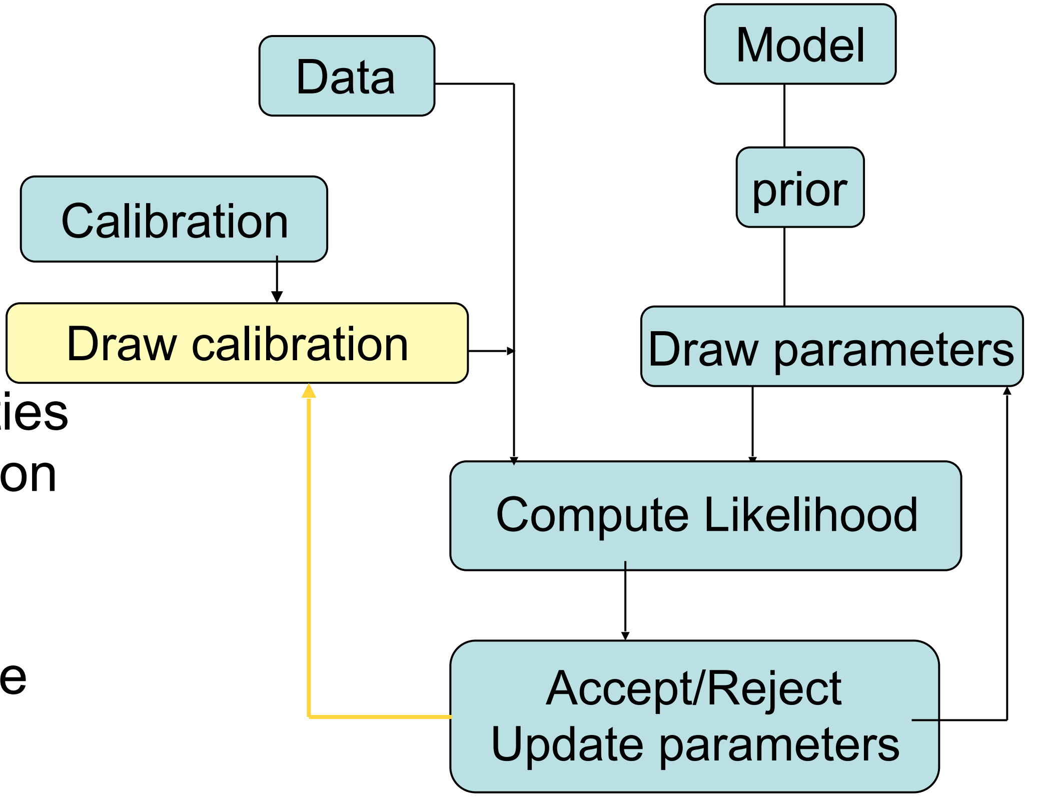
Sherpa, MCMC and Bayesian Analysis

MCMC samplers:

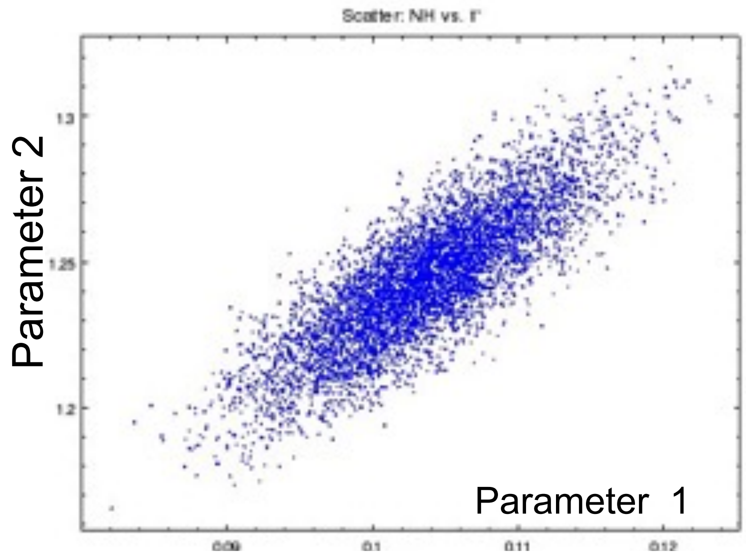
Metropolis and Metropolis-Hastings algorithms

Support for Bayesian analysis with priors.

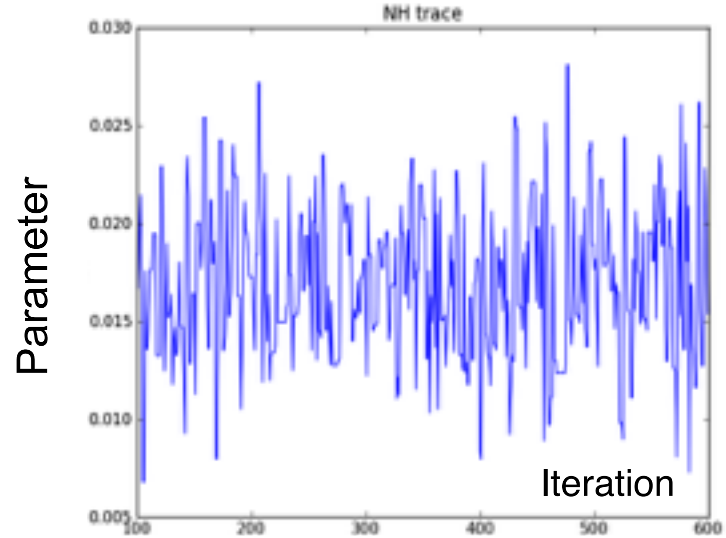
- Explores parameter space and summarizes the full posterior or profile posterior distributions.
- Computed parameter uncertainties can include systematic or calibration errors.
- Simulates replicate data from the posterior predictive distributions.

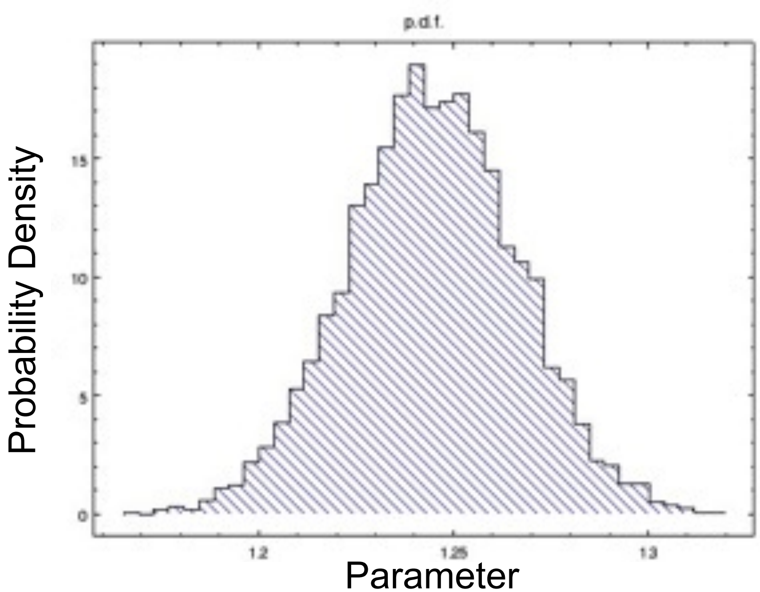


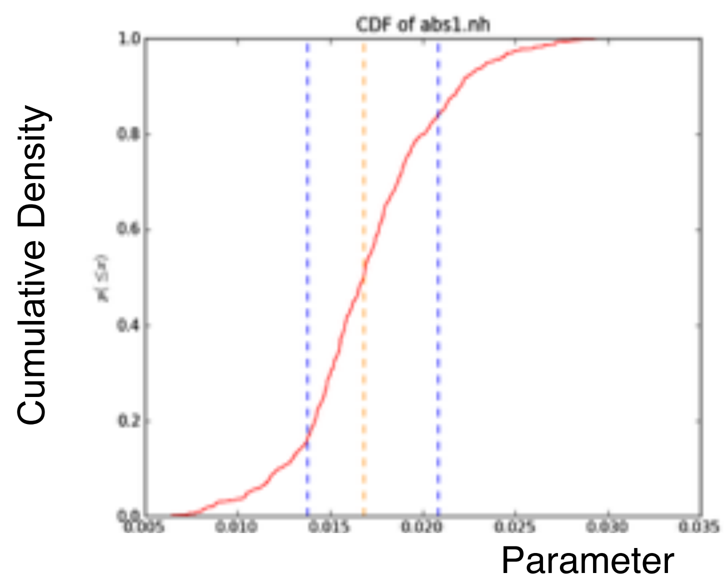
Visualization of the MCMC Results



Trace of a parameter during MCMC run







Confidence Limits

Essential issue = after the bets-fit parameters are found estimate the confidence limits for them. The region of confidence is given by (Avni 1976):

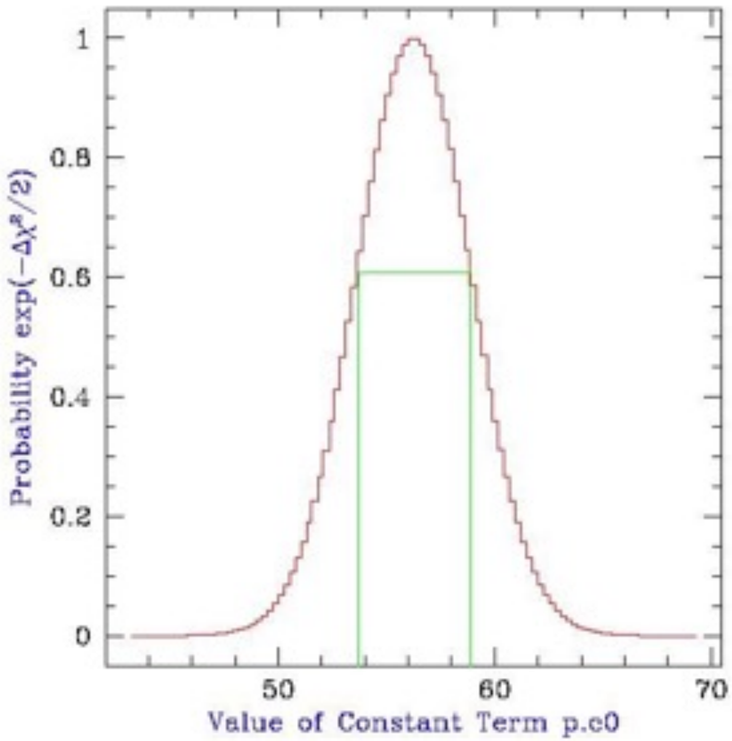
$$\chi^2_{\alpha} = \chi^2_{min} + \Delta(\nu,\alpha)$$

ν - degrees of freedom
α - level
χ^2_{min} - minimum

Δ depends only on the number of parameters involved not on goodness of fit

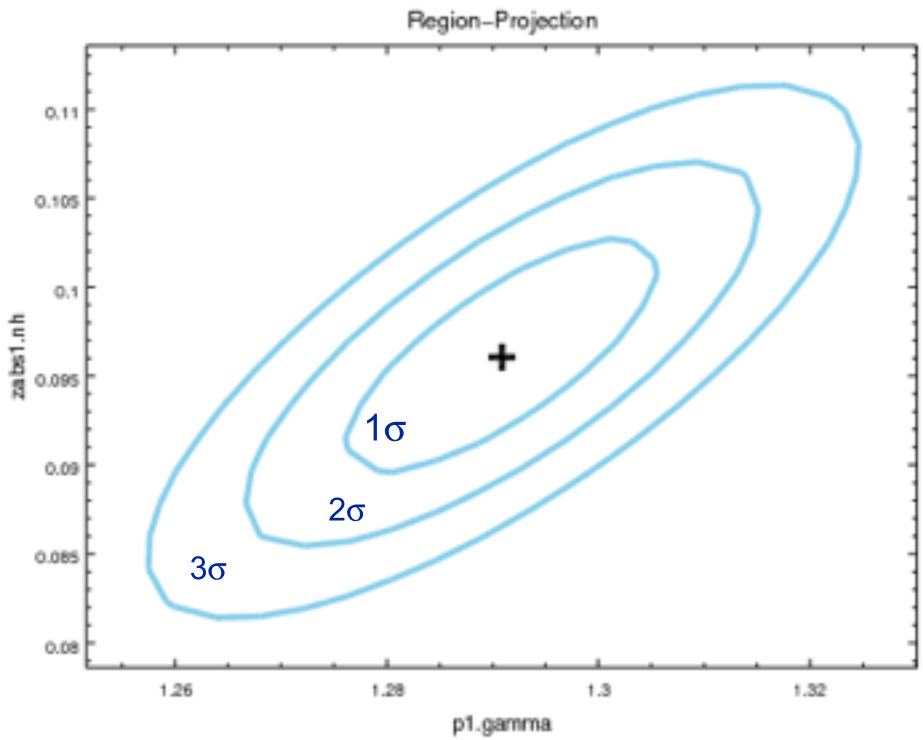
TABLE 1
CONSTANTS FOR CALCULATING CONFIDENCE REGIONS

α (%)	q (No. of Interesting Parameters) 1	2	3
68.........	1.00	2.30	3.50
90.........	2.71	4.61	6.25
99.........	6.63	9.21	11.30



Confidence Regions

```
sherpa-61>
reg_proj(p1.gamma,zabs1.nh,nloop=[20,20])
sherpa-62> print get_reg_proj()
min     = [ 1.2516146   0.07861824]
max     = [ 1.33010494  0.11357147]
nloop   = [20, 20]
fac     = 4
delv    = None
log     = [False False]
sigma   = (1, 2, 3)
parval0 = 1.29085977295
parval1 = 0.0960948525609
levels  = [ 634.40162888  638.28595426  643.93503803]
```



Examples of Sherpa Science Cases

Spatial Fitting of the TeV emission in H.E.S.S. observations

A&A 541, A5 (2012)

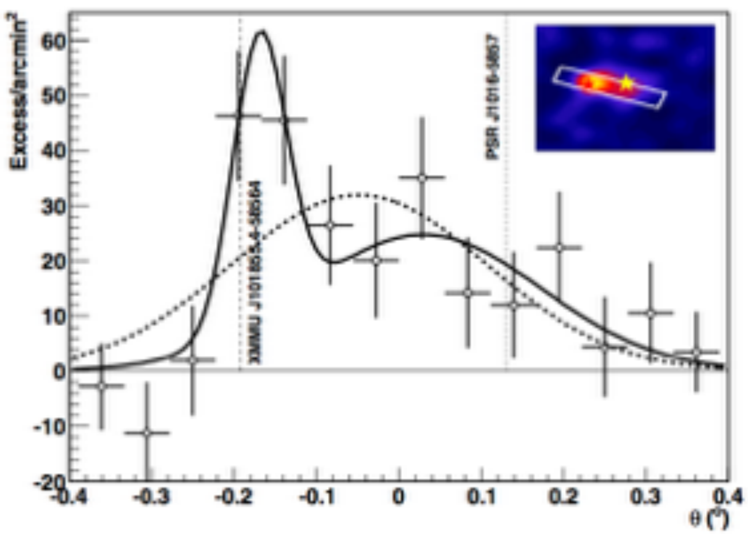


Fig. 3. Profile of the VHE emission along the line between the peak of the point-like emission and the peak of the diffuse emission, as illustrated in the inset. Fits using a single and a double Gaussian function are shown in dashed and solid lines respectively. The positions of XMMU J101855.4–58564 and PSR J1016–5857 are marked with dashed and dotted vertical lines and red and yellow stars in the inset, in which the significance image obtained using an oversampling radius of 0.1° is shown.

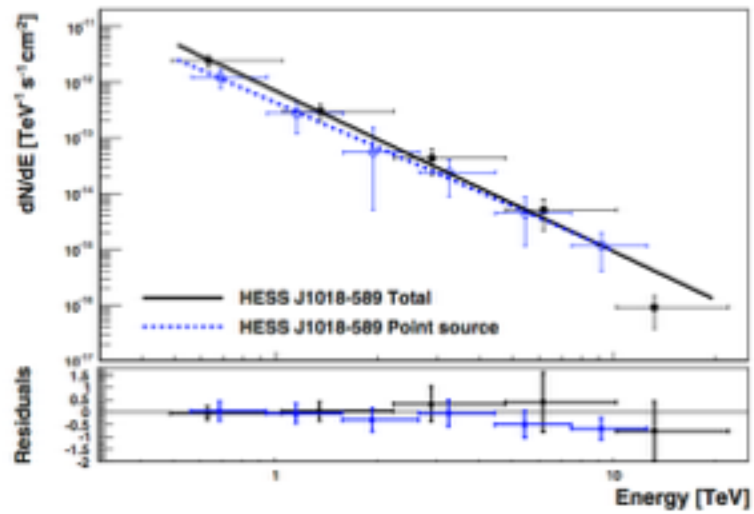
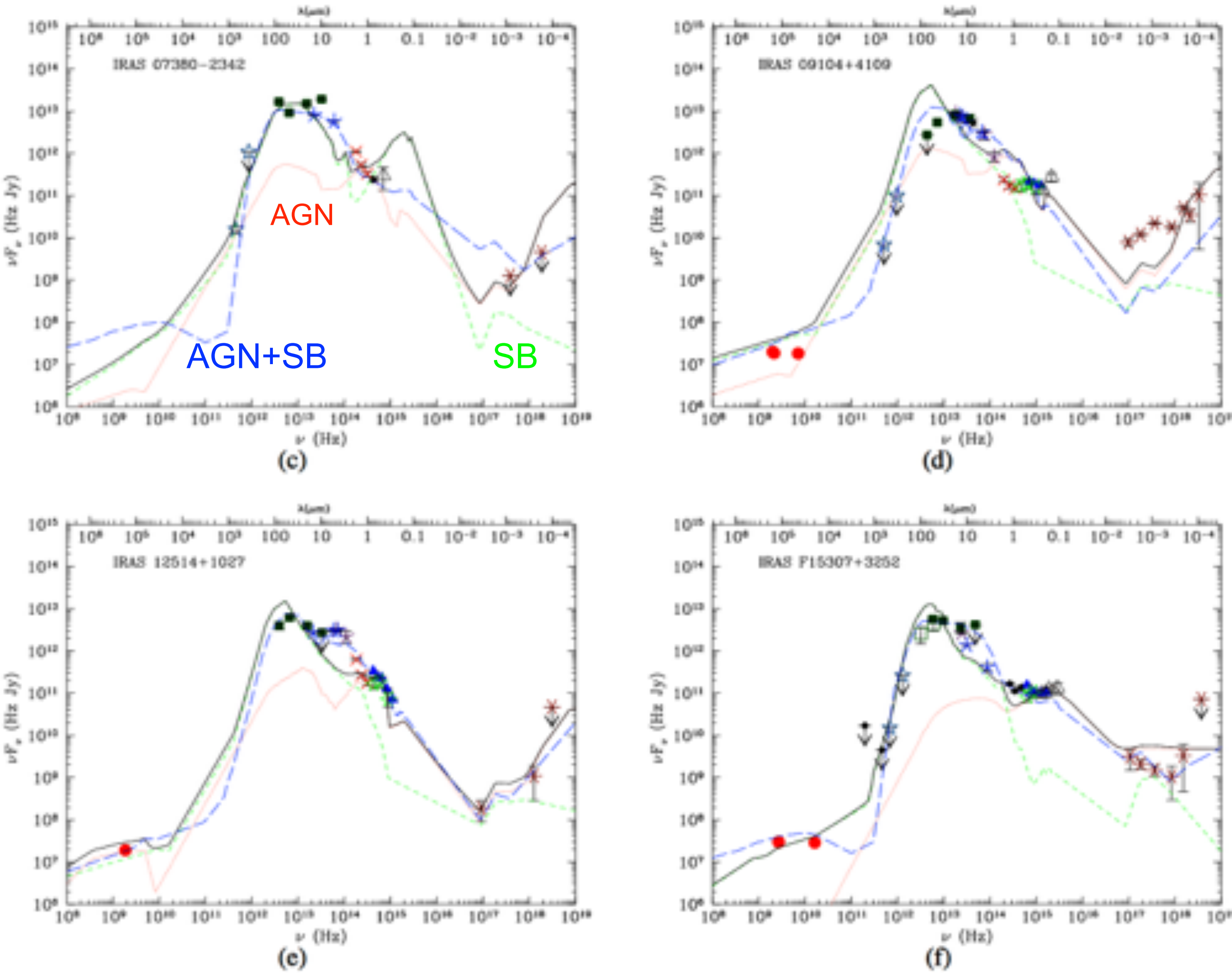


Fig. 4. VHE photon spectrum of HESS J1018–589 for a point-like source at position A (in blue dots and dashed blue line) and derived from a region of size 0.30° comprising the point-like and diffuse emission (in black dots and solid black line). The residuals to the fit are shown in the bottom panel.

Abramowski et al. (2012)

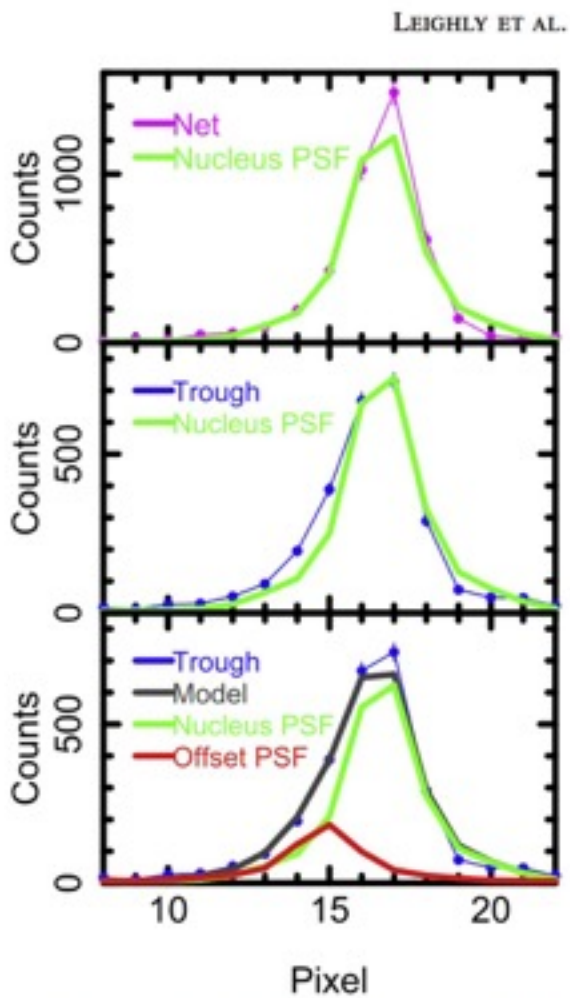
Spectral (SED) Fitting with Composite Templates

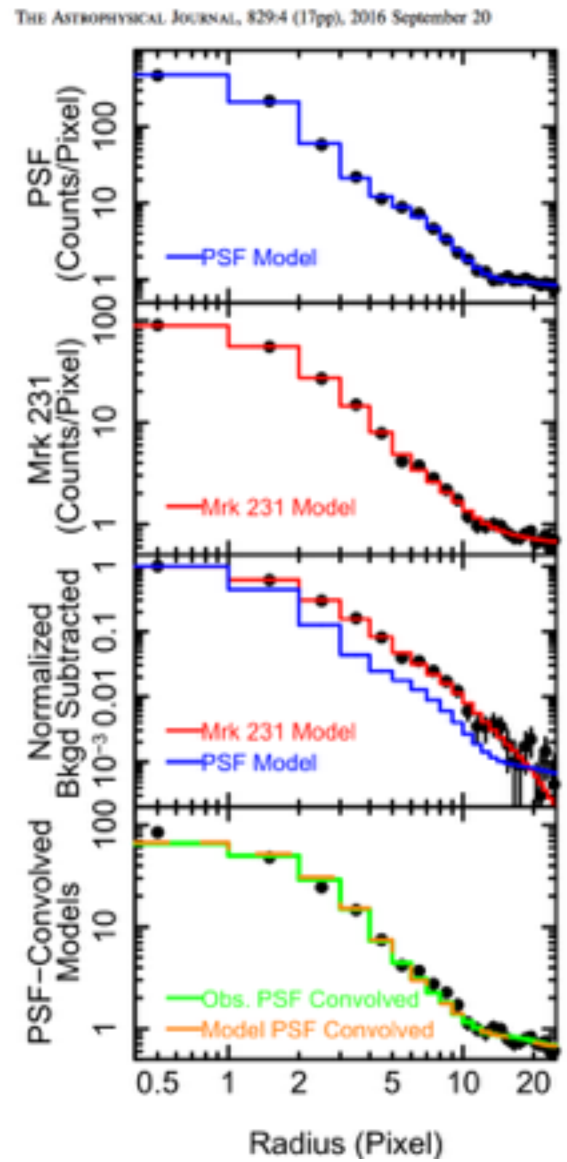


Ruiz et al. (2010)

Fig. 6. Rest-frame SED of class B HLIRG and their best-fit models. Symbols as in Fig. 5. The long-dashed lines (blue in the colour version) are the best fits obtained using composite templates (see Sects. 4.1 and 5.2).

Fitting Spatial Profiles of the HST observations of Mrk 231

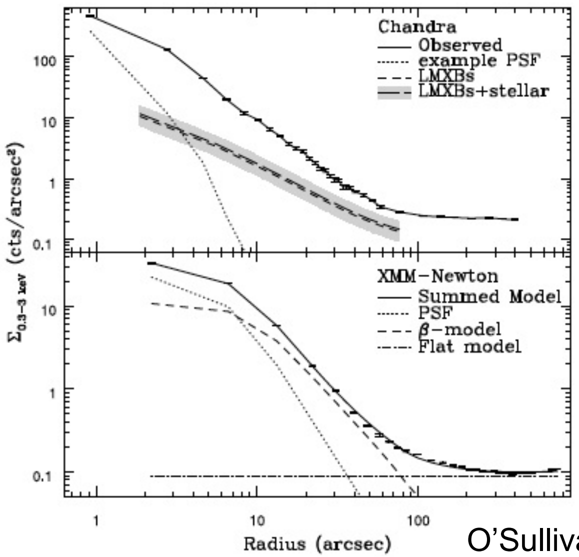




Leighly et al. (2016)

Surface Brightness Profiles
(with & without PSF)

Chandra and XMM



O'Sullivan et al. (2011)

HST Images

Radio loudness and surface brightness profile 2167

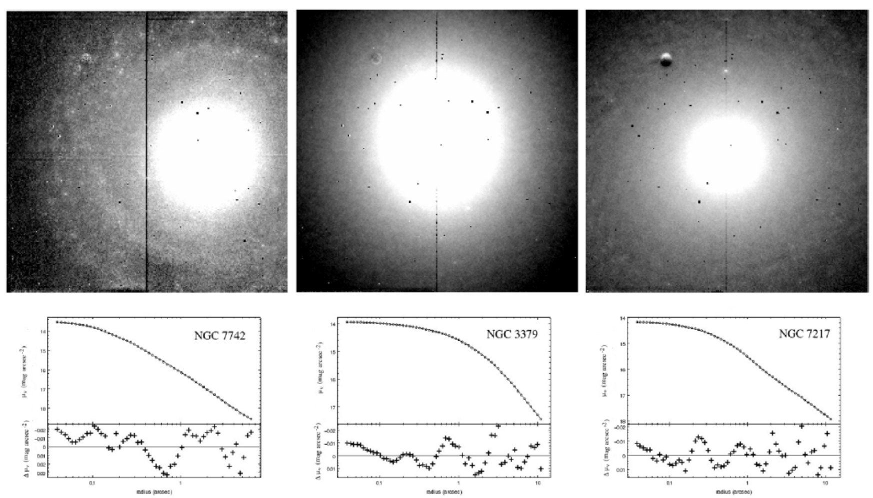
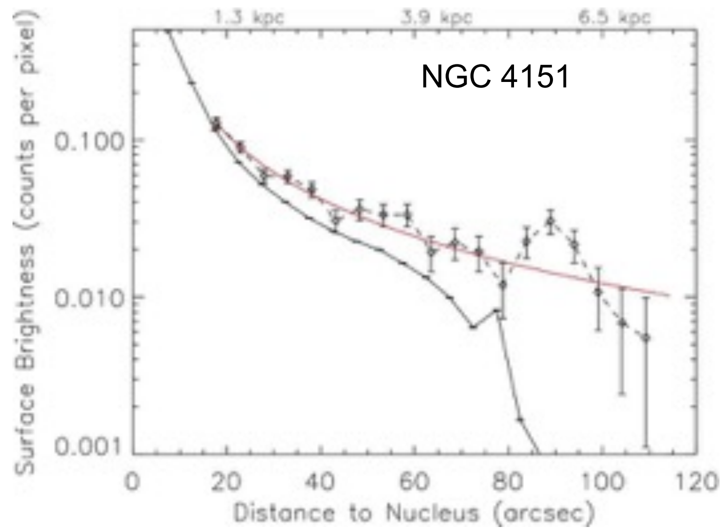


Figure 1. Galaxy images (top row) and radial brightness profiles (bottom row) for a confident Sérsic fit (NGC 7742; left), Core fit (NGC 3379; centre) and Double-Sérsic fit (NGC 7217; right).

Richings, Utley & Kording (2011)

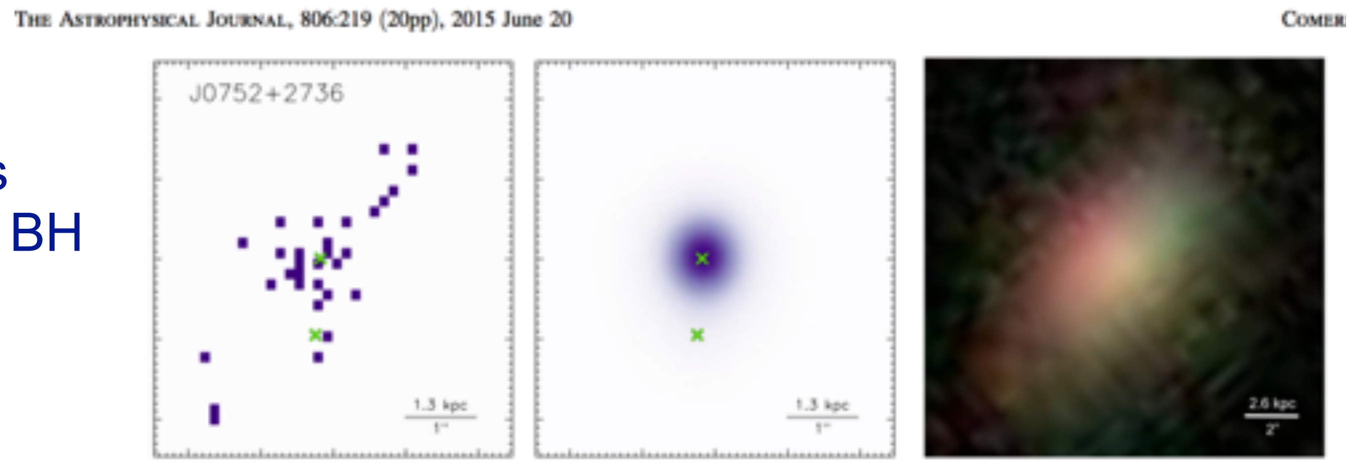
Chandra



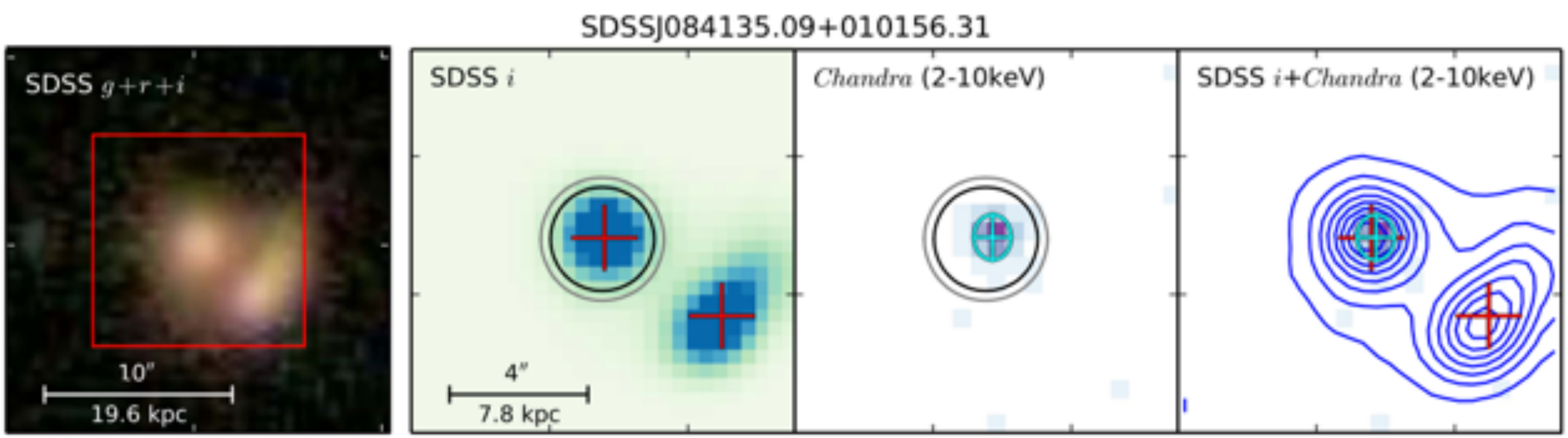
Wang et al. (2010)

Image Analysis

Optical-X-ray offsets
Searches for Binary BH and GW Recoils

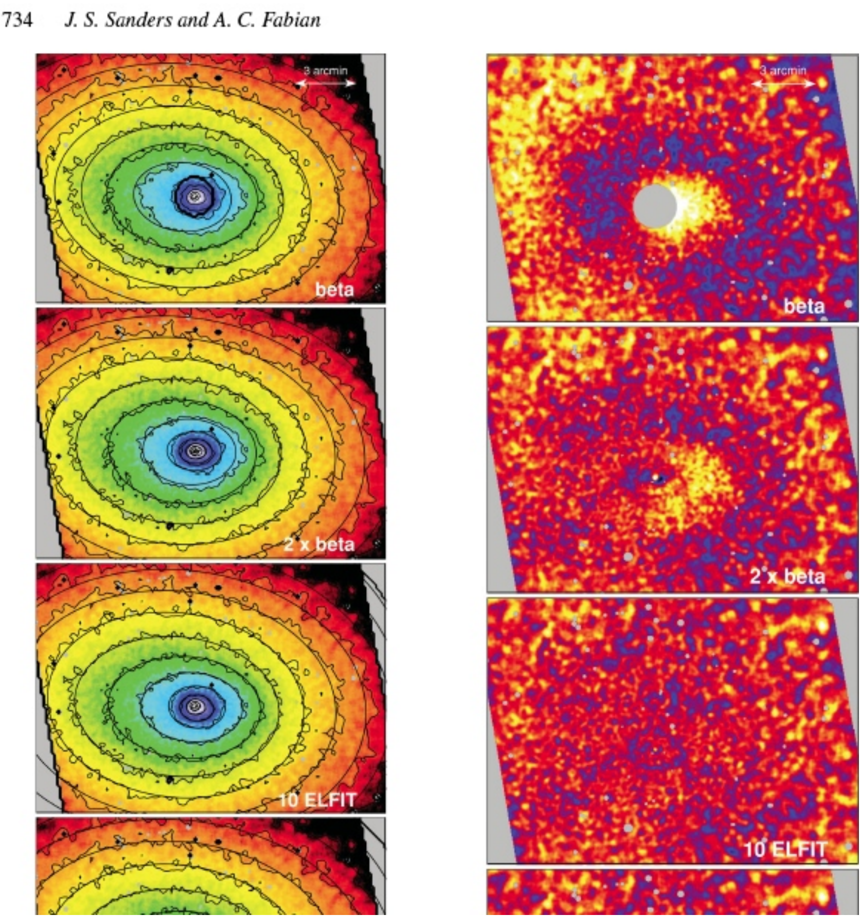


Comerford et al. (2015)

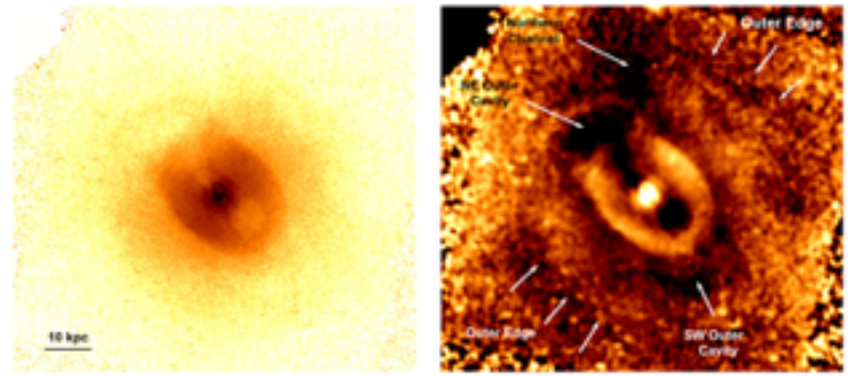


Barrows et al. (2016)

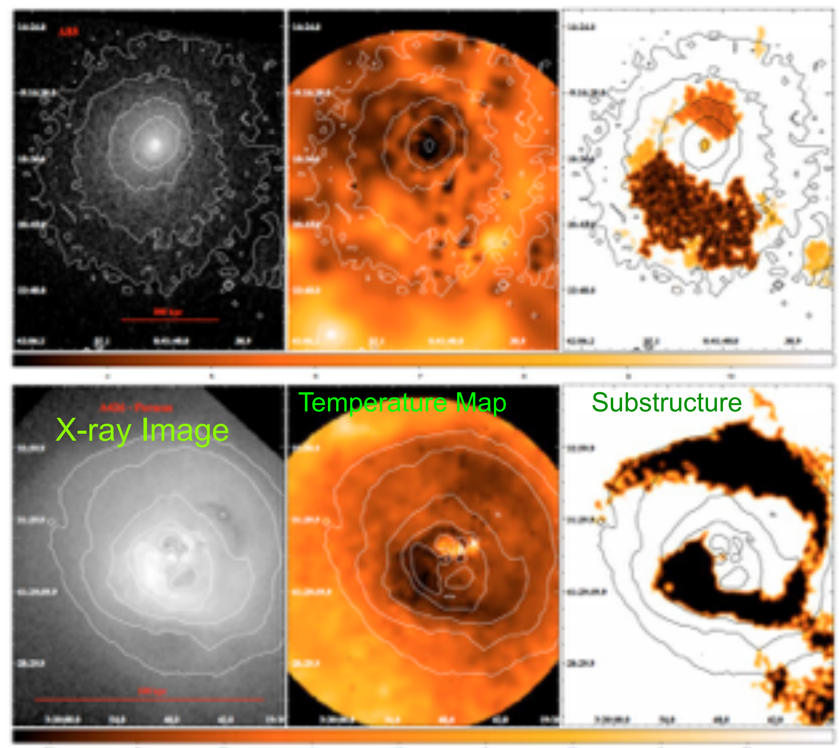
Identifying Substructures in X-ray Clusters



Sanders & Fabian (2012)



Randall et al. (2015)



Lagana, Santos & Lima Neto (2010)

Sherpa CIAO 4.11 Web Pages

http://cxc.harvard.edu/sherpa

Freeman, P., Doe, S., & Siemiginowska, A. 2001, SPIE 4477, 76

Doe, S., et al. 2007, Astronomical Data Analysis Software and Systems XVI, 376, 543

Refsdal et al. 2009 - Sherpa: 1D/2D modeling and fitting in Python in Proceedings of the 8th Python in Science conference (SciPy 2009), G Varoquaux, S van der Walt, J Millman (Eds.), pp. 51-57

https://sherpa.readthedocs.io/en/4.11.0/

Sherpa
4.11.0

Search docs

INTRODUCTION

- Installation
- A quick guide to modeling and fitting in Sherpa
- Sherpa and CIAO

USER DOCUMENTATION

- What data is to be fit?
- Creating model instances
- Evaluating a model
- Available Models
- What statistic is to be used?
- Optimisers: How to improve the current parameter values
- Fitting the data
- Visualisation
- Markov Chain Monte Carlo and Poisson data
- Utility routines

WORKED EXAMPLES

- Simple Interpolation
- Simple user model

AN INTERACTIVE APPLICATION

- Using Sessions to manage models and

Docs » Welcome to Sherpa's documentation

Edit on GitHub

Sherpa

Welcome to the Sherpa documentation. Sherpa is a Python package for modeling and fitting data. It was originally developed by the Chandra X-ray Center for use in analysing X-ray data (both spectral and imaging) from the Chandra X-ray telescope, but it is designed to be a general-purpose package, which can be enhanced with domain-specific tasks (such as X-ray Astronomy). Sherpa contains an expressive and powerful modeling language, coupled with a range of statistics and robust optimisers.

Sherpa is released under the GNU General Public License v3.0, and is compatible with Python versions 2.7, 3.5, 3.6, and 3.7. Information on recent releases and citation information for Sherpa is available using the Digital Object Identifier (DOI) 10.5281/zenodo.593753.

Introduction

- Installation
  - Quick overview
  - Requirements
  - Releases and version numbers
  - Installing a pre-compiled version of Sherpa
  - Building from source
  - Testing the Sherpa installation
- A quick guide to modeling and fitting in Sherpa
  - Getting started
  - Fitting a one-dimensional data set

Summary

- Sherpa is a Python package.
- It provides models, fit statistics and optimization methods for variety of problems in astronomy.
- It is flexible and extensible as it accepts new models, statistics or optimization.
- Sherpa can also be included in a modeling software in Python.