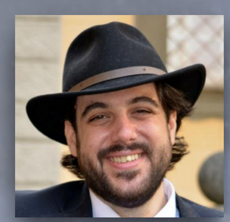
Sherpa
On the Move to Open, Collaborative Development



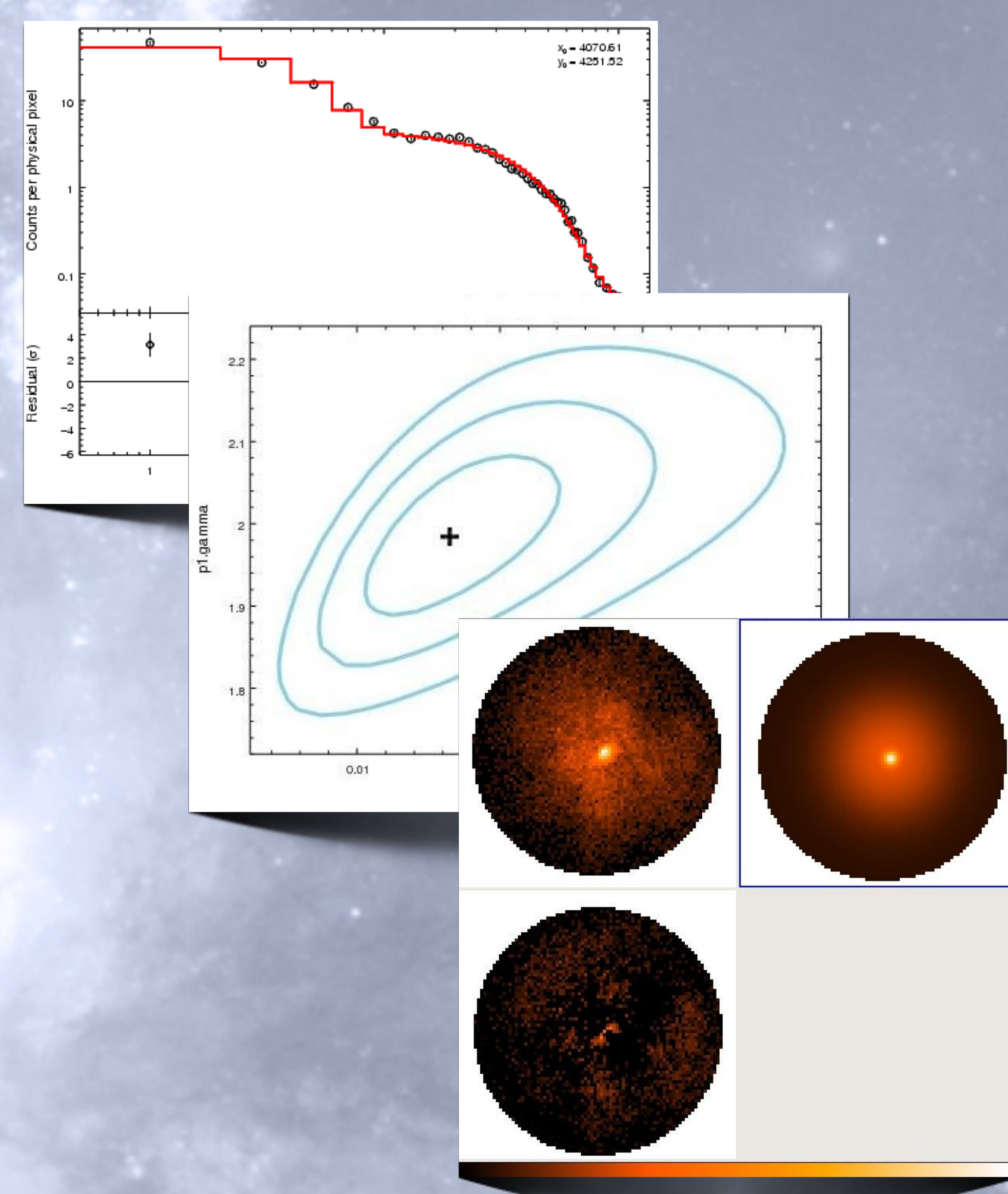
O. Laurino, A. Siemiginowska, J. Evans, T. Aldcroft, D. J. Burke, J. McDowell, W. McLaughlin, D. Nguyen
olaurino@cfa.harvard.edu

Sherpa is the Chandra Interactive Analysis of Observations (CIAO) modeling and fitting application. Written in Python, with efficient C, C++, and Fortran extensions, Sherpa enables the user to construct complex models from simple definitions and fit those models to data, using a variety of statistics and optimization methods.

Sherpa is a general-purpose fitting engine with advanced capabilities, and has been used as a backend for the development of new applications like Iris, the Virtual Astronomical Observatory spectral energy distribution builder and analyzer. However, building and installing Sherpa as a standalone Python package was problematic, and such a build would not maintain all of the Sherpa capabilities.

For version 4.7 Sherpa's build scripts have been completely rewritten, standardized, and made independent of CIAO, so that Sherpa can now be built as a fully functional standalone Python package, and yet allow users the flexibility they need in order to build Sherpa in customized environments.

Customized source build options example:
- Link Sherpa's Python extensions against local libraries, e.g. FFTW
- Enable XSPEC extension for X-Ray specific models (HEASARC)



Sherpa enables you to:
- fit 1-D data sets (simultaneously or individually), including: spectra, surface brightness profiles, light curves, general ASCII arrays;
- fit 2-D images/surfaces in the Poisson/Gaussian regime;
- access the internal data arrays;
- build complex model expressions;
- import and use your own models;
- choose appropriate statistics for modeling Poisson or Gaussian data;
- import new statistics, with priors if required by analysis;
- visualize a parameter space with simulations or using 1-D/2-D cuts of the parameter space;
- calculate confidence levels on the best-fit model parameters;
- choose a robust optimization method for the fit: Levenberg-Marquardt, Nelder-Mead Simplex or Monte Carlo/Differential Evolution;
- perform Bayesian analysis with Poisson Likelihood and priors, using Metropolis or Metropolis-Hastings algorithm in the MCMC (Markov-Chain Monte Carlo);
- use Python to create complex analysis and modeling functions, build the batch mode analysis or extend the provided functionality to meet the required needs.

1. python setup.py install

2. pip install [--pre] sherpa

The "pre" switch is required as Sherpa is currently tagged as a pre-release on PyPI

3. conda install sherpa

The CXC channel currently needs to be added with
$ conda config --add https://conda.binstar.org/cxc

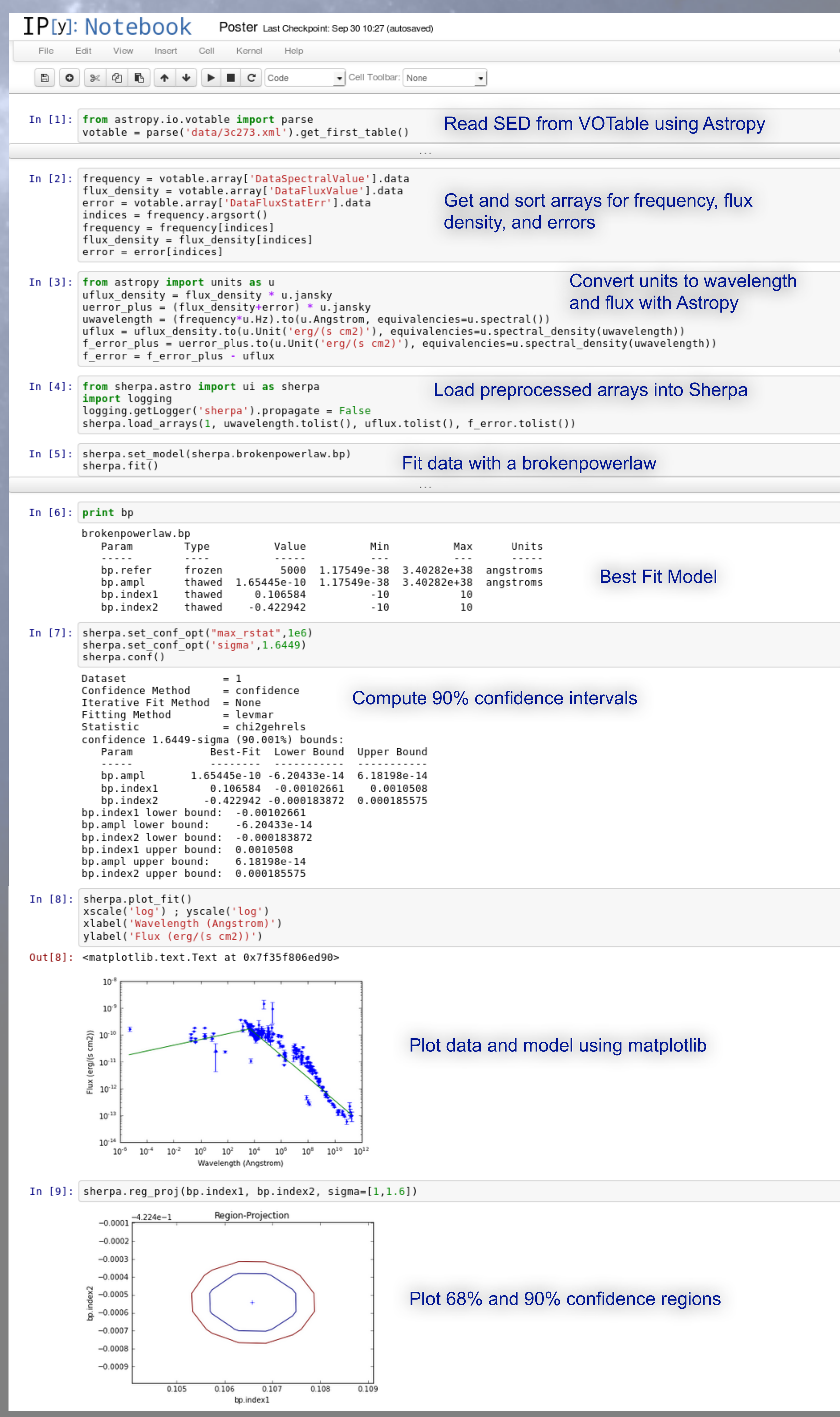
4. bash sherpa-…-installer.sh

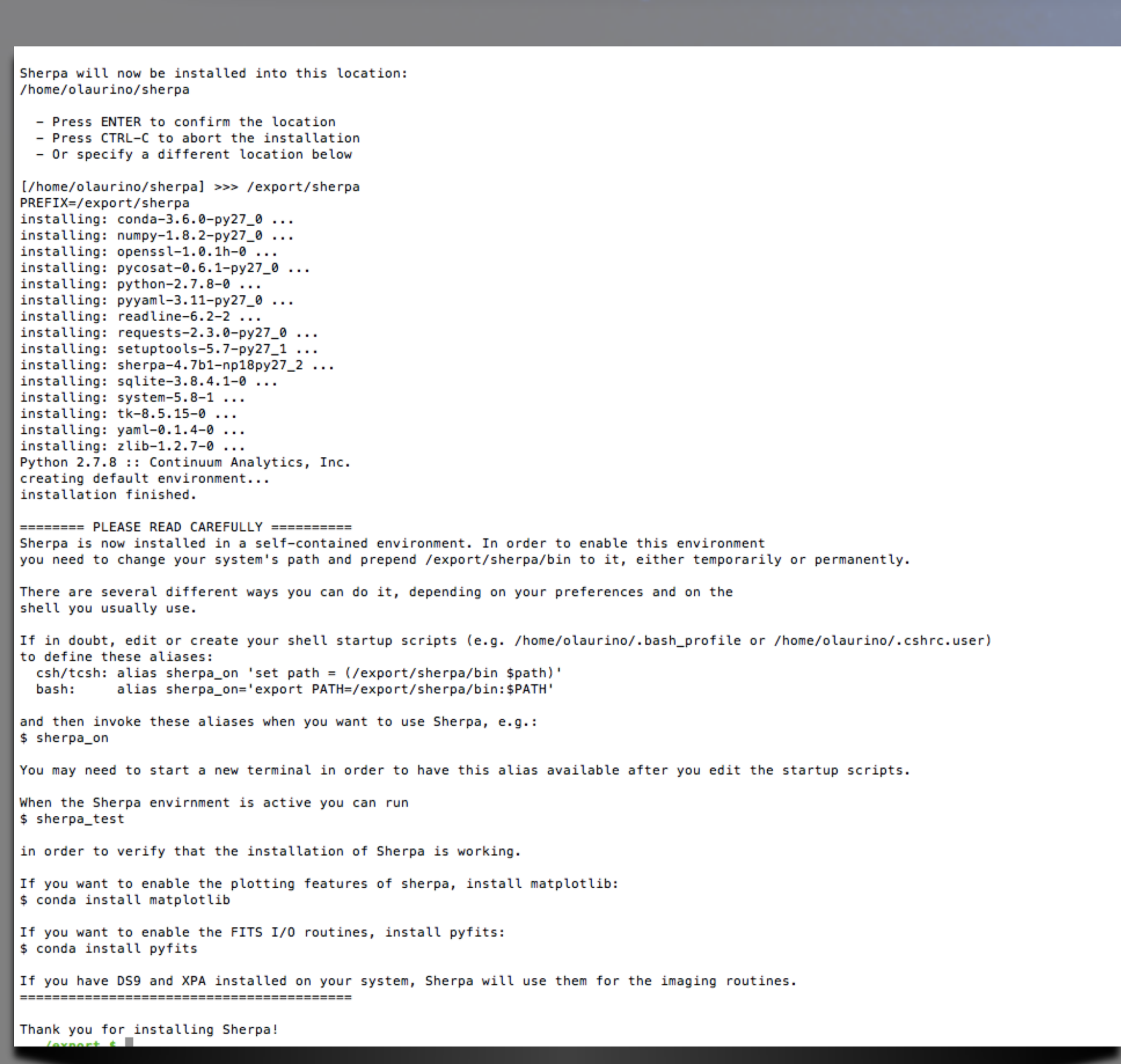
Sherpa can be seamlessly integrated with other Python tools and packages.

In the example on the right, Sherpa is used alongside Astropy to perform a simple fit in an IPython Notebook.

Note how Sherpa's plot_fit() function can be used to produce an inline matplotlib plot.

Sherpa supports PyFITS and Matplotlib as FITS and plotting backends, as well as Crates and ChIPS, which are the native CIAO packages for FITS I/O and plotting.





```
Sherpa will now be installed into this location:
/home/olaurino/sherpa

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/olaurino/sherpa] >>> /export/sherpa
PREFIX=/export/sherpa
installing: conda-3.6.0-py27_0 ...
installing: numpy-1.8.2-py27_0 ...
installing: openssl-1.0.1h-0 ...
installing: pycosat-0.6.1-py27_0 ...
installing: python-2.7.8-0 ...
installing: pyyaml-3.11-py27_0 ...
installing: readline-6.2-2 ...
installing: requests-2.3.0-py27_0 ...
installing: setuptools-5.7-py27_1 ...
installing: sherpa-4.7b1-np18py27_2 ...
installing: sqlite-3.8.4.1-0 ...
installing: system-5.8-1 ...
installing: tk-8.5.15-0 ...
installing: yaml-0.1.4-0 ...
installing: zlib-1.2.7-0 ...
Python 2.7.8 :: Continuum Analytics, Inc.
creating default environment...
installation finished.

======== PLEASE READ CAREFULLY ==========
Sherpa is now installed in a self-contained environment. In order to enable this environment
you need to change your system's path and prepend /export/sherpa/bin to it, either temporarily or permanently.

There are several different ways you can do it, depending on your preferences and on the
shell you usually use.

If in doubt, edit or create your shell startup scripts (e.g. /home/olaurino/.bash_profile or /home/olaurino/.cshrc.user)
to define these aliases:
  csh/tcsh: alias sherpa_on 'set path = (/export/sherpa/bin $path)'
  bash:     alias sherpa_on='export PATH=/export/sherpa/bin:$PATH'

and then invoke these aliases when you want to use Sherpa, e.g.:
$ sherpa_on

You may need to start a new terminal in order to have this alias available after you edit the startup scripts.

When the Sherpa envirnment is active you can run
$ sherpa_test

in order to verify that the installation of Sherpa is working.

If you want to enable the plotting features of sherpa, install matplotlib:
$ conda install matplotlib

If you want to enable the FITS I/O routines, install pyfits:
$ conda install pyfits

If you have DS9 and XPA installed on your system, Sherpa will use them for the imaging routines.
==========================================

Thank you for installing Sherpa!
```

Sherpa is available in source and binary form and can be easily installed with:
1. setuptools, using the source distribution
2. pip, as Sherpa is registered in PyPI
3. conda, from an Anaconda installation
4. standalone installer

The figure on the left shows the output of the standalone installer.

Beta Binaries can be downloaded from:
http://cxc.cfa.harvard.edu/contrib/sherpa47b
(or scan the QR Code on the right).

Production-ready binaries and full documentation for the source builds will be released with CIAO 4.7 in December 2014.

Support for the development of Sherpa is provided by the National Aeronautic and Space Administration through the Chandra X-ray Center, which is operated by the Smithsonian Astrophysical Observatory for and on behalf of NASA under contract NAS8-03060.

This work is part of a larger framework that includes migrating the whole Sherpa codebase to GitHub, so that the wider astronomical community can be engaged in its development, as well as providing developers with a clean software framework for extending Sherpa.