

CIAO in 2015-16: Work Behind the Scenes Plus Scripts, PSF and Sherpa Python Package

Antonella Fruscione and Douglas Burke
for the CIAO team

The demand for CIAO (the *Chandra* Data Analysis software) remains steady since 2011 with about 100-150 downloads per month, and has been stronger than ever since the CIAO 4.8 release in December 2015. *Chandra* and non-*Chandra* users continue to take advantage of the growing capabilities that CIAO has to offer and of the extensive documentation which accompany them!

CIAO 4.8—released in Dec 2015 and patched in February 2016—is the latest software release and was primarily a maintenance release to address needed upgrades behind the scenes: bug fixes, support of new compilers and upgrade of off-the-shelf software used in the system.

This release also includes additional support for the analysis of observations in *Continuous Clocking* (CC) mode that directly affects users: both the *acis_process_events* and the *tg_resolve_events* tool have been updated to allow for better calibration of ACIS+HETG data taken in CC mode. The *chandra_repro* script has also been upgraded to take advantage of these changes.

With every major CIAO release a large effort goes into updating all the relevant documentation on the website and in the tools' help files to ensure that any software change is properly reflected in the text. Furthermore, at release time and during the year, new documents are written not only to support and explain changes in software, but also in response to questions or comments from the community.

The *HETG/ACIS-S/CC-Mode Grating Spectra thread*, published in Dec. 2015, belongs to the first category and was newly created to help users working with ACIS/HETG observations in CC mode. The new “Why topic” *Pitfalls using PIMMS for Observed Data* and the FAQ entry *What does “zero length polygon line segment” warning mean?* are two recent documents belonging to the second category.

The “Why topic” highlights the issues that arise from using *PIMMS, the Portable Interactive Multi-Mission Simulator*, and the *CXC proposal planning counterpart* to estimate fluxes from observed count rates. Specifically, the report addresses the fact that for bright

sources and especially for large surveys the systematic errors introduced by using PIMMS may become significant and introduce a bias into the results. The new FAQ addresses in detail the reason for a common warning seen while running CIAO tools.

The appearance and the behaviour of the main CIAO download page (the first entry point for the majority of users) has been restructured in CIAO 4.8, with a clearer and more prominent “standard install” button and more precise custom installation options for users with specific needs. The goal is first and foremost to make any part of CIAO, including the installation, as easy as possible for users.

CIAO scripts

In the spirit of making CIAO more accessible to all users—including users who are not X-ray astronomy specialists—we have continued the development and improvement of the high level programs (the *CIAO “scripts”*), which have the goal of simplifying the most common types of analysis.

Several scripts were released in the past months, but three are especially worth noting here: *download_obsid_caldb*, *install_marx*, and *simulate_psf*.

Several HelpDesk tickets have reported problems downloading CALDB due to size (3 Gb zipped, 7.5 Gb unzipped). In reality only a small fraction of the *Chandra* CALDB is needed to analyze any given ObsID.

The *download_obsid_caldb* script determines which CALDB files are necessary for a given ObsID and downloads only those missing from the user's existing

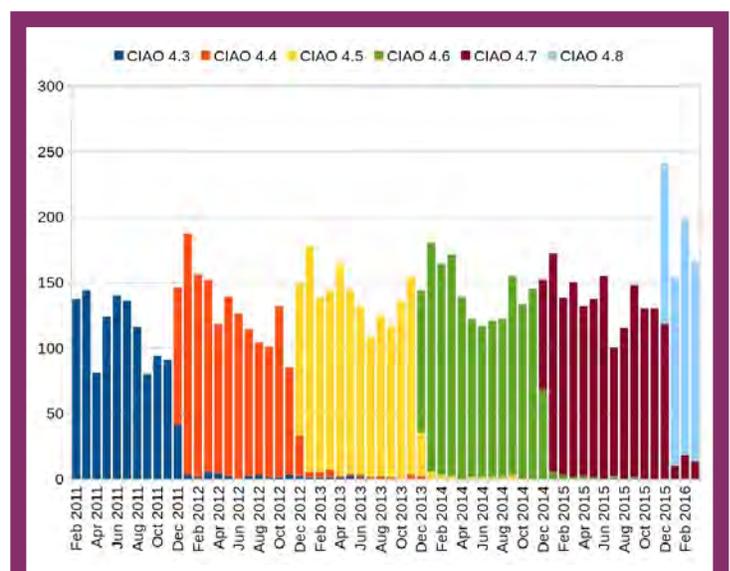
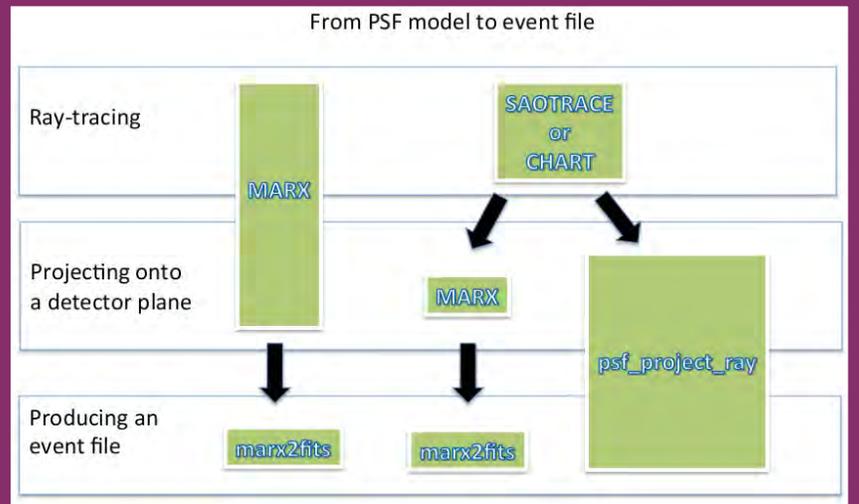


Figure 1: CIAO monthly download statistics since 2011

Figure 2: A schematic view of the PSF modeling paths: while MARX can be used for both ray tracing and projecting onto a detector-specific plane, the rays produced by SAOTrace or ChaRT require an extra step to project the rays onto a detector-specific plane (via MARX) or onto a semi-infinite detector-plane (via `psf_project_ray`). The `psf_project_ray` tool produces an events file as output, while the `marx2fits` routine should be used after running a MARX simulation to obtain an event file. The `simulate_psf` script currently covers the steps on the left column (MARX + `marx2fits`) and can project and produce an event file if rayfiles from ChaRT are given as input.



CALDB. The files are added to user's standard CIAO CALDB location and users can accumulate CALDB files as needed, ObsID by ObsID. It is particularly helpful for users with less-than-fully-reliable internet connections where downloading a multi-GB file can time out. For example, using the new [download_obsid_caldb](#) script for one ObsID will decrease the size of the necessary download to ~0.3 GB, whereas downloading the entire CALDB would have required ~7.5GB. If the user also needs background files, [download_obsid_caldb](#) would result in ~0.35 GB instead of the 13 GB required to download the entire CALDB with background files.

The two other scripts aim to simplify the simulation of the *Chandra* PSF:

The [install_marx](#) script automates the standard steps necessary to download, build, and install [MARX](#), (the *Chandra* on-orbit simulator), while [simulate_psf](#) wraps the complexity of the thread in which MARX is used to simulate an existing observation. The script's design and interface will allow other simulators in the future (e.g. [SAOTrace](#)) and currently accepts rayfiles generated by [ChaRT](#) (the CXC web interface to SAOTrace) as input. Both scripts are part of the major effort to collect and disseminate information about the *Chandra* PSF described below.

PSF Central

The documentation project called "[PSF Central](#)" represents an effort to unify under one umbrella the wealth of information about the *Chandra* PSF (dictionary, ahelp files, threads, Why documents etc.) that was previously scattered around the CIAO webpages. The PSF Central website is currently organized in a

tool-centric direction (what tools exist related to the *Chandra* PSF) but future plans are to extend it toward a more science-centric direction (e.g., is my source extended? is this jet real? etc.).

Within PSF Central we endeavor to better explain the relationship between SAOTrace, ChaRT, and MARX while simplifying installation and use of these applications.

The two scripts mentioned above, together with a new release of ChaRT that allows for simulations compatible with sub-pixel analysis, are instrumental in achieving this goal.

Several science threads have also been updated or written anew to clarify the use of ChaRT and MARX to simulate existing or planned observations.

Sherpa Python Fitting

While Sherpa (the *Chandra* modelling and fitting application) is an integral part of CIAO, a close reading of last year's "15 years in CIAO" *Chandra* newsletter article (Fruscione et al. 2015) will have revealed our parallel project of moving it to the code-development site GitHub so that its development is open to everyone. In this environment, Sherpa can be used for general fitting and can be built independently of the CIAO infrastructure. The project can be found at <https://github.com/sherpa/sherpa/>. The first major release from this work was Sherpa 4.8.0, which is available as a part of the CIAO 4.8 release, as well as directly from GitHub. The newest release, Sherpa 4.8.1, occurred in April 2016.

The primary aims of these releases are to make Sherpa easier to include in other Python projects and to allow users to import the Sherpa module into a Python

session. Sherpa is available via the Sherpa Anaconda channel for those who use the Anaconda distribution, as well as directly from GitHub. The plan is to have several minor releases throughout 2016, leading up to a major release in December, which will be coordinated with the next CIAO release. The individual Sherpa releases are also archived by the Zenodo Science Archive, thanks to its integration with GitHub. As an example, the details of the Sherpa 4.8.0 release are available at <https://zenodo.org/record/45243>.

As befitting our move to a world in which everything is tracked, we now have a number of helpful visualizations to show how development is progressing, such as this “Punch Card” view (hopefully this will not provide painful memories for those readers “lucky” enough to have programmed with actual punch cards):

It is unclear, at this time, whether the lunch talk held by the High-Energy Division here at the Center for Astrophysics can explain the drop off in Sherpa work around mid-day Wednesday. More data is obviously needed!

The preferred installation location for Sherpa is within the Anaconda Python environment. To install Sherpa into a separate environment (called “sherpa” in this example), one could use the following:

```
% conda config --add channels
https://conda.anaconda.org/sherpa
% conda create -n=sherpa python=2.7
sherpa matplotlib astropy ipython
% source activate sherpa
```

Additional installation methods are available, for example, to also build the optional XSPEC model interface, and are described on the standalone Sherpa page at <http://cxc.cfa.harvard.edu/contrib/sherpa/>.

For anyone interested in using Sherpa in their own project, or adding new functionality, please head on over to <https://github.com/sherpa/sherpa/>, download the code, and start coding! Development and integration of Sherpa with Astropy will go forward during the summer through the work of a student participating in the “Google Summer of Code” project. ■

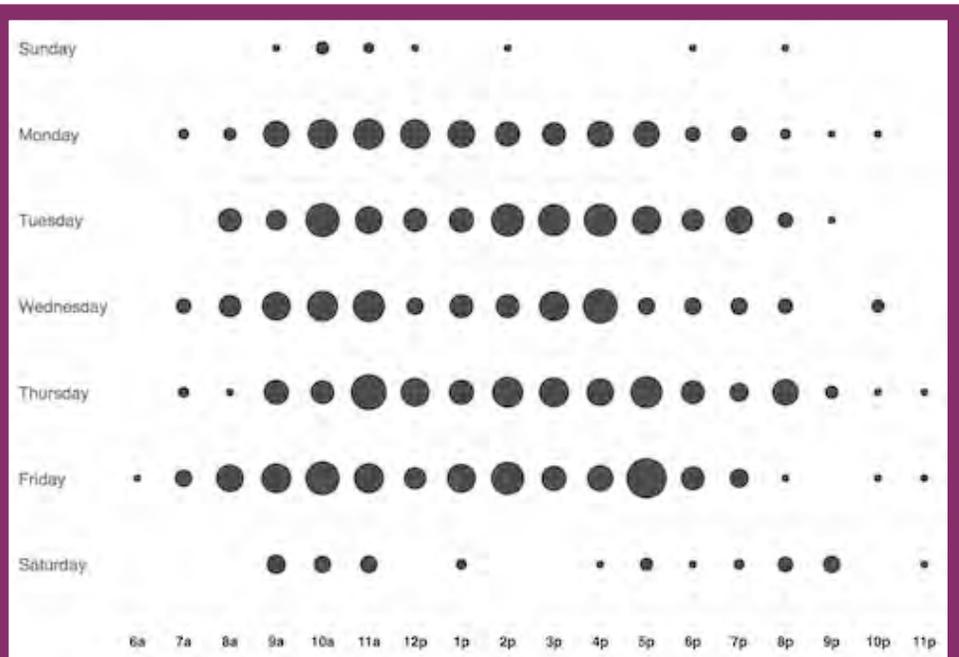


Figure 3: “Punch Card” view from Sherpa GitHub

How to stay up-to-date about CIAO and Sherpa

Several channels are available to keep abreast of CIAO and any new development or problems:

-  [@ChandraCIAO](https://twitter.com/ChandraCIAO)
-  [ChandraCIAO](https://www.facebook.com/ChandraCIAO)
-  [+ChandraCIAO](https://plus.google.com/+ChandraCIAO)

- Subscribe to the *Chandra* Electronic Announcements at <http://cxc.harvard.edu/cdo/udb/userdat.html>
- Check “What’s New” <http://cxc.harvard.edu/ciao/news.html> or subscribe to the CIAO News RSS feed at <http://cxc.harvard.edu/ciao/feed.xml>
- Check the YouTube channel “4ciaodemos” at <http://www.youtube.com/user/4ciaodemos>

Questions to the Sherpa team at the CXC can be made either through the [GitHub web site](#) or via the [CXC HelpDesk](#).

	SAOTrace	ChaRT	MARX	arccorr
user interface	command-line	web interface to SAOTrace	command-line	command-line
how to obtain?	download → compile → install locally	web-based, no installation required	download → compile → install locally or via install_marx	CIAO tool
how many sources?	multiple sources	individual point sources	multiple sources	individual point sources
current limitations	does not run on the latest Linux and Mac platforms	limited number of available parameters (but sufficient for typical use cases)	does not include all of the fine HRMA details	only mono-chromatic sources supported
pros	flexible and powerful; definitive HRMA model	easy to use	high fidelity, and yet easy to use; portable; includes models for HRMA, gratings, and detectors	simple, quick tool
cons	complex installation and usage; only includes the HRMA model	need to wait for output	not as high-fidelity as SAOTrace for HRMA	circular approximation of the PSF

Figure 4: A quick comparison between the different modeling tools available to Chandra users

CIAO Workshop - 15 August 2016

The CXC is soliciting interest in a one-day CIAO workshop on August 15, 2016 just prior to the start of the *Chandra Science for the Next Decade* workshop. Depending on the level of interest, the workshop may be extended into the morning of August 16. The CIAO workshop will be tailored to the attendees but usually consists of a few talks in the morning with ample time in the afternoon for hands-on sessions with CXC experts to assist individuals with their specific data analysis questions.

Possible topics for the talks include:

- Introduction to X-ray astronomy
- Advanced CIAO
- Topics in *Chandra* Calibration
- Intro or Advanced Sherpa
- Intro or Advanced MARX
- Intro or Advanced ds9

Users interested in attending the CIAO Workshop can find more information at:

<http://cxc.harvard.edu/ciao/workshop/index.html>



An example Sherpa run through

Once installed in the python environment, Sherpa can be imported like any python module. In the following example, we demonstrate fitting of a low-order polynomial to randomized data (this example is based on the “Quick Start” IPython notebook guide provided with Sherpa, which can be found at <http://nbviewer.jupyter.org/github/sherpa/sherpa/blob/master/docs/SherpaQuickStart.ipynb>). The warning message lets us know that the version of Sherpa being used does not include the XSPEC models interface, and the --classic and --no-banner flags are used to simplify the screen output of IPython:

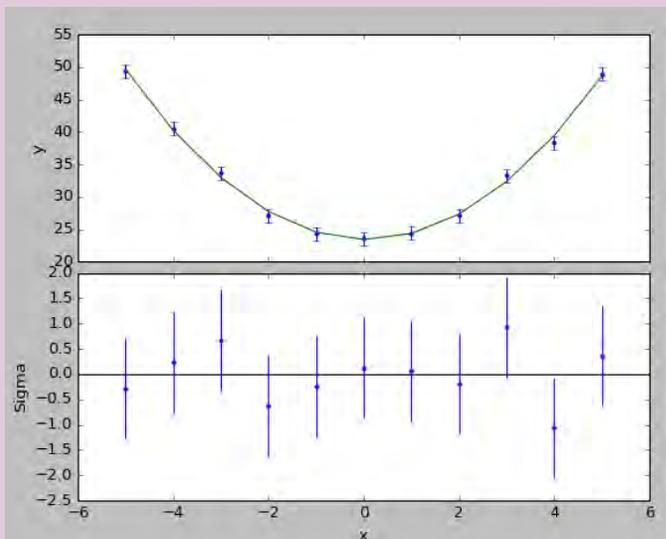
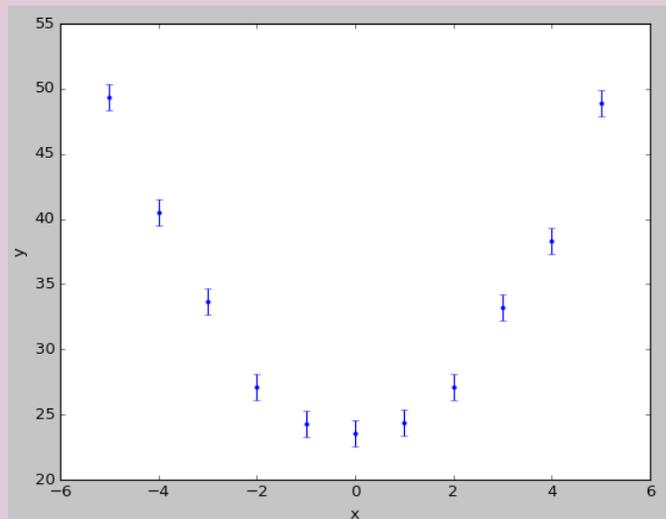
```
% ipython --classic --no-banner --pylab

Using matplotlib backend: Qt4Agg
>>> from sherpa.astro import ui
WARNING: failed to import sherpa.astro.xspec;
XSPEC models will not be available
>>> import numpy as np
>>> x = np.arange(-5, 5.1)
>>> y = x*x + 23.2 +
      np.random.normal(size=x.size)
>>> e = np.ones(x.size)
>>> ui.load_arrays(1, x, y, e)
>>> ui.plot_data()
>>> ui.set_model(ui.polynom1d.poly)
>>> print(poly)

polynom1d.poly
  Param   Type          Value          Min
  ----   -
poly.c0   thawed          1 -3.40282e+38  3.40282e+38
poly.c1   frozen           0 -3.40282e+38  3.40282e+38
poly.c2   frozen           0 -3.40282e+38  3.40282e+38
poly.c3   frozen           0 -3.40282e+38  3.40282e+38
poly.c4   frozen           0 -3.40282e+38  3.40282e+38
poly.c5   frozen           0 -3.40282e+38  3.40282e+38
poly.c6   frozen           0 -3.40282e+38  3.40282e+38
poly.c7   frozen           0 -3.40282e+38  3.40282e+38
poly.c8   frozen           0 -3.40282e+38  3.40282e+38
poly.offsetfrozen  0 -3.40282e+38  3.40282e+38

>>> ui.thaw(poly.c1, poly.c2)
>>> ui.fit()
Dataset           = 1
Method            = levmar
Statistic         = chi2
Initial fit statistic = 12653.3
Final fit statistic = 3.22478 at function evaluation 8
Data points      = 11
Degrees of freedom = 8
Probability [Q-value] = 0.919469
Reduced statistic = 0.403098
Change in statistic = 12650.1
  poly.c0      23.4186
  poly.c1     -0.107496
  poly.c2      1.02587
>>> ui.plot_fit_delchi()

# see figure to the right
```



```
>>> ui.conf()
poly.c0 lower bound:  -0.455477
poly.c0 upper bound:  0.455477
poly.c2 lower bound:  -0.0341394
poly.c1 lower bound:  -0.0953463
poly.c2 upper bound:  0.0341394
poly.c1 upper bound:  0.0953463
Dataset                = 1
Confidence Method      = confidence
Iterative Fit Method   = None
Fitting Method         = levmar
Statistic               = chi2gehrels
confidence 1-sigma (68.2689%) bounds:
  Param   Best-Fit   Lower Bound   Upper Bound
  -----
poly.c0   23.4186   -0.455477    0.455477
poly.c1   -0.107496  -0.0953463   0.0953463
poly.c2   1.02587   -0.0341394   0.0341394
>>> ui.reg_proj(poly.c0, poly.c2, nloop=[21, 21])
# see figure to the right
```

