

- Yaxx Quick-start Guide
 - Install Yaxx
 - Enter CIAO environment
 - Create Analysis Root Directory
 - Copy Input Chandra Data
 - Create Object List File
 - Copy Configuration Files
 - Run Yaxx
 - Inspect Key Analysis Data Files
 - Perform Detailed Interactive Fitting
 - Publish Results
-

Yaxx Quick-start Guide

This document describes the basic steps in setting up and running **yaxx** to perform X-ray spectral analysis of a Chandra data. It combines a step-by-step tutorial with discussion of some of the most common user options. An in-depth description of **yaxx** is given in the User Manual.

Install Yaxx

The installation and basic testing of **yaxx** is covered in the installation documentation. Throughout this document the variable `$YAXX` is assumed to be the directory in which **yaxx** was installed. For instance, if you install **yaxx** in your home directory as shown in the installation guide, you would do (for `cs`h or `tcsh`):

```
set YAXX=~ /yaxx
```

Enter CIAO environment

Source the appropriate initialization script (as instructed in the CIAO documentation) to enter the CIAO analysis environment. For `cs`h or `tcsh` do

```
source <CIAO_HOME>/bin/ciao.csh
```

Note that if the `FTOOLS` package is used within the same session as CIAO then the `FTOOLS` initialization must be done *before* CIAO initialization. Search for “Other Software Packages” in the CIAO documentation for further details.

Create Analysis Root Directory

Yaxx is run from an analysis root directory that should be distinct from the `$YAXX` source directory. For instance

```
mkdir ~/yaxx_analysis
cd ~/yaxx_analysis
```

Copy Input Chandra Data

Spectral analysis of Chandra data with **yaxx** requires ACIS event data and aspect solution data, both of which are found in the primary data products for an ObsId. These input data can be stored either outside the analysis directory (e.g. in a raw data repository) or within the analysis directory in a structure that meshes with the output spectral analysis results. Here we demonstrate the latter, using data provided in the **yaxx** installation.

```
mkdir obs3102
cp $YAXX/Test/Data/obs3102/acisf03102_evt2.fits obs3102/
cp $YAXX/Test/Data/obs3102/pcadf_asol1.fits obs3102/
```

The directory location of both the input Chandra data and the output analysis results are configurable, with the caveat that the outputs are stored in sub-directories named *obs<ObsId>>*.

Create Object List File

The list of ObsIds and sources, along with other source-specific information such as position and redshift, is specified as tabular data in the object list file. The format of this file can be FITS, RDB, or ASCII (with several common column delimiters supported).

To create a object list file, create a new file named *sample.dat* and insert the following lines:

obsid	src	redshift	X	Y	object
3102	1	0.32	4167	4085	Q1250+568

This indicates that just one source named Q1250+568 with physical sky coordinates (4167,4085) will be extracted. Any fits that rely on a redshift will use 0.32.

Note that the output source list from *celldetect* or *wavdetect* can be used to provide the source location (X,Y), as described in the reference manual.

Copy Configuration Files

The operation of **yaxx** is controlled by configuration files named *yaxx.cfg*. **Yaxx** looks for configuration files in four directories in the following order:

\$YAXX/yaxx.cfg

System-wide configuration data. In general this should not be modified. This file includes the file naming specifications, basic sherpa fitting scripts, etc.

yaxx.cfg (in analysis root directory)

Configuration data for each yaxx analysis ‘project’. This file typically includes basic run parameters, the fit models and rules, and report formatting rules, but any item in the system-wide yaxx.cfg file can also be specified.

`<output_dir>/obs<obsid>/yaxx.cfg [optional]`

Configuration data specific to a particular obsid. If processing a number of ObsIds this allows specification of different run parameters for one ObsId.

`<output_dir>/obs<obsid>/src<src>/yaxx.cfg [optional]`

Configuration data specific to a particular source. For example one source may require a different fit method to converge properly. One could also create a source-specific fit model with particular starting fit parameters that help with convergence. In large batch-fitting applications there are occasionally oddballs that benefit from fine tuning.

For the tutorial example, copy the **yaxx** installation `User/yaxx.cfg` file into the analysis directory:

```
cp $YAXX/User/yaxx.cfg ./yaxx.cfg
```

Examine this file and note some of the key options available such as the input data and output directories, the object list file, fit methods and statistics, source extraction parameters, fit rules, and model definitions.

Details for all parameters are given in the **yaxx** manual

Run Yaxx

From your analysis root directory run **yaxx** by entering

```
$YAXX/yaxx
```

If you have installed the `yaxx` launcher script somewhere in your search path you can simply run

```
yaxx
```

You will then see **yaxx** processing ObsId 3102. At the end of a processing run **yaxx** will declare SUCCESS if all steps succeeded. The key processing steps are:

- **Start a log file**
- **Copy or link input data to analysis output directory**
- **Make source and background extraction region files**
- **Extract spectrum (PI or PHA file) from event data**
- **Make image of source event data**
- **Fit specified models to spectral data with *sherpa***
- **Make report summary pages (postscript and HTML)**

Each processing step is run only if the output files for that step are non-existent or older than the input files. The typical way to force reprocessing from a particular step is to tell **yaxx** to delete certain files before processing using the `-preclean` option. For instance, to re-do the spectral fitting:

```
yaxx -preclean fit
```

Since the report summary depends on the fit results, it will also be regenerated.

Assuming the processing is successful, examine the results with

```
firefox report_index.html
```

where `firefox` can be replaced by the name of your favorite web browser. This shows the **yaxx** report index which links to report pages for the individual sources. The report page shows a summary of the source parameters, an image of the source and the extraction regions used, a table of spectral fit results, and plots of the spectral model fits. In this tutorial there is only one source, but if there are multiple sources then individual reports are linked together by the arrows in the upper left corner of each report.

A latex/postscript report is also created and can be viewed with:

```
gv obs3102/src1/report.ps
```

Inspect Key Analysis Data Files

The analysis data files associated with `yaxx` processing have the structure:

```
<output_dir>/obs<obsid>/src<src>
```

The files unique to an ObsId are within the `obs<obsid>>` directory, and those unique to the particular source are in the `src<src>>` (i.e. the “source” directory). Some of the more useful files in the source directory are:

```
acis*.pi      : Various pi (or pha) spectral files for fitting
acis.rmf      : RMF
acis.arf      : ARF
<fit_model>.in : Sherpa script used to fit <fit_model>, e.g. pl.in
report.html   : Final fit summary report (HTML)
report.ps     : Final fit summary report (postscript)
report.tex    : Final fit summary report (latex)
log           : Processing log. log.<n> files are old logs
```

Perform Detailed Interactive Fitting

The `<fit_model.in>>` file is a very useful starting point for doing more detailed or interactive spectral fitting of a particular source. For the simple powerlaw model in this example one can exactly recreate the **yaxx** fitting steps in *sherpa* by doing:

```
cd obs3102/src1
sherpa
use pl.in
```

Publish Results

The LaTeX report summary *report.tex* within the source directory contains a table of fit results which can be easily copied into a paper for publication. We request that a citation be given if you use **yaxx** in your analysis. A detailed citation for **yaxx** is given in the reference manual.