

Yaxx XMM Quick-start Guide

Table of Contents

- 1 Overview
- 2 Configure shell environment
- 3 Create Analysis Root Directory
- 4 Copy Input XMM Data
- 5 Create Object List File
- 6 Copy Configuration Files
- 7 Run Yaxx
- 8 Inspect Key Analysis Data Files
- 9 Adjust source and background regions
- 10 Perform Detailed Interactive Fitting

1 Overview

This document describes the basic steps in configuring and running *yaxx* to perform X-ray spectral analysis of *XMM* data. It combines a step-by-step tutorial with discussion of some of the most common user options. It is assumed that *yaxx* has been successfully installed and tested as described in the installation guide. This includes the important step of setting the correct CIAO, SAS, and HEADAS environment initialization commands.

The *yaxx XMM* processing thread is based on the More-or_less Complete Example in the XMM-NEWTON ABC GUIDE. Contributions to the thread or suggestions from users with expertise in *XMM* data analysis would be greatly welcome.

2 Configure shell environment

The different environments used in X-ray analysis by have the potential to conflict if not initialized in the correct order (e.g. CIAO and FTOOLS). To minimize problems, *yaxx* does the analysis environment initialization internally (in the correct order) and expects that the user has *not* done any initialization.

To proceed with this quick start example, ensure that you have a "clean" environment by opening a new terminal window. If your shell initialization script (e.g. `~/ .cshrc`) automatically does environment setup (e.g. FTOOLS) then you may experience problems running *yaxx*.

Next set the `$YAXX` variable to point to the *yaxx* installation directory. For instance, if you installed *yaxx* in your home directory as shown in the installation guide, you would do (for `csh` or `tcsh`):

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

3 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
```

4 Copy Input XMM Data

Spectral analysis of XMM data with *yaxx* requires starting from the ODF data for an ObsId. These input data must be stored within the analysis directory in a structure that meshes with the output spectral analysis results.

For this tutorial, download the ODF data for ObsId 0147511201 from the XMM Newton Archive and move it into the analysis directory:

```
mkdir 0147511201
mv /path/to/xmm_data/0147511201/ODF 0147511201/
```

Unlike with Chandra, the directory location of the input XMM data and the output analysis results are not fully configurable.

5 Create Object List File

The list of X-ray sources to be processed by *yaxx* is specified as tabular data in the object list file. Each table row contains data such as the source ObsId and position. The format of this file can be FITS, RDB, or ASCII (with several common column delimiters supported).

To create an object list file for this example, create a new file named `sample.dat` and insert the following lines:

obsid	src	redshift	ra	dec	rad	object	orbit
0147511201	1	0.3	163.58804	57.42903	600	"Mrk ABC"	0527

This indicates that just one source with the (bogus) name "Mrk ABC" with celestial coordinates (163.58804,57.42903) will be extracted. Any fits that rely on a redshift will use 0.3. The `obsid` and `orbit` must correspond to the XMM Obsid and Orbit (including any leading zeroes). The source extraction radius is 600 pixels (30 arcsec). This value corresponds to the default if no extraction radius (`rad` column) is supplied.

6 Copy Configuration Files

The operation of *yaxx* is controlled by configuration files named `yaxx.cfg`. The key configuration files are:

System: `$YAXX/yaxx.cfg`

System-wide configuration data. In general this should not be modified after installation.

Thread: `$YAXX/resources/<thread>/yaxx.cfg`

Configuration data for each *yaxx* analysis "thread", examples of which include *Chandra* and *XMM*. These files define the specific processing steps, environments, and file naming conventions for each thread.

Project: `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. Any items in the system or thread `yaxx.cfg` files can be overridden by the project configuration file.

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

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

Examine this file and note some of the options specified such as the input data and output directories, mission and thread, 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* reference manual.

7 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 0147511201. At the end of the processing run (a couple of hours later) *yaxx* will declare SUCCESS if all steps succeeded.

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 C<-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.

8 Inspect Key Analysis Data Files

The analysis data files associated with *yaxx* processing in this example have the structure:

```
<output_dir>/<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:

```
mos1*           : MOS-1 files (event files for source, PI, RMF, ARF, background, etc)
mos2*           : MOS-2 files
pn*             : PN files
<fit_model>.in : Sherpa script used to fit <fit_model>, e.g. pl.in
<fit_model>.*  : Other files associated with <fit_model>
report.html    : Final fit summary report (HTML)
log            : Processing log. log.<n> files are old logs
```

9 Adjust source and background regions

The default source and background regions generated by *yaxx* can be adjusted by doing:

```
cd 0147511201/src1
emacs mos1_src.reg    # Or your favorite editor
emacs mos1_bkg.reg
```

The source and background region files are named `{mos1,mos2,pn}_{src,bkg}.reg`.

Once the region files have been edited appropriately simply re-run *yaxx*. During processing *yaxx* will reprocess only steps that depend on the region files. *Yaxx* will never overwrite an existing source or background region file, so one can also manually create the region files before starting *yaxx*. This requires making the files in the expected output directories.

10 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 0147511201/src1
sherpa
use pl.in
```