Not just an X-ray Extractor

Tom Aldcroft, SAO/CXC
http://cxc.harvard.edu/contrib/yaxx

Yaxx is a Perl script that facilitates batch data processing using Perl open source tools and commonly available astronomical software (CIAO/*Sherpa*, SAS, HEAsoft). For X-ray analysis it includes automated spectral extraction, fitting, and report generation. The primary emphasis is on having a simple tool that can be run without requiring an extensive learning curve. However, for those with the motivation, *yaxx* is highly configurable and can be customized to support complex pipeline data analysis. In particular the *yaxx* processing flow is fully configurable, easily allowing automation of data reduction steps. *Yaxx* includes default processing threads and output templates for *Chandra* and *XMM* spectral analysis.

Key Features:

- Uses Perl and CIAO/*Sherpa*, plus other packages (e.g. SAS and FTOOLS) as needed
- Full reference manual, quick-start guide, and installation guide
- Easily customized via configuration and template files
- Takes advantage of the powerful *Sherpa* and *S-lang* scripting capability
- Uses file dependencies in each processing step for selective reprocessing
- Processing summaries created in HTML and postscript formats
- Files and *Sherpa* fit script ready for interactive analysis
- Generates convenient FITS tables of all spectral fitting results
- Completely free, as in beer and open source software
- Supported on Linux and Solaris platforms

Yaxx is Simple

Fit one source with default models

Copy input data

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

Create 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
```

Copy standard configuration file

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

Run *yaxx*

```
$YAXX/yaxx
```

View fit results

```
firefox report_index.html
```

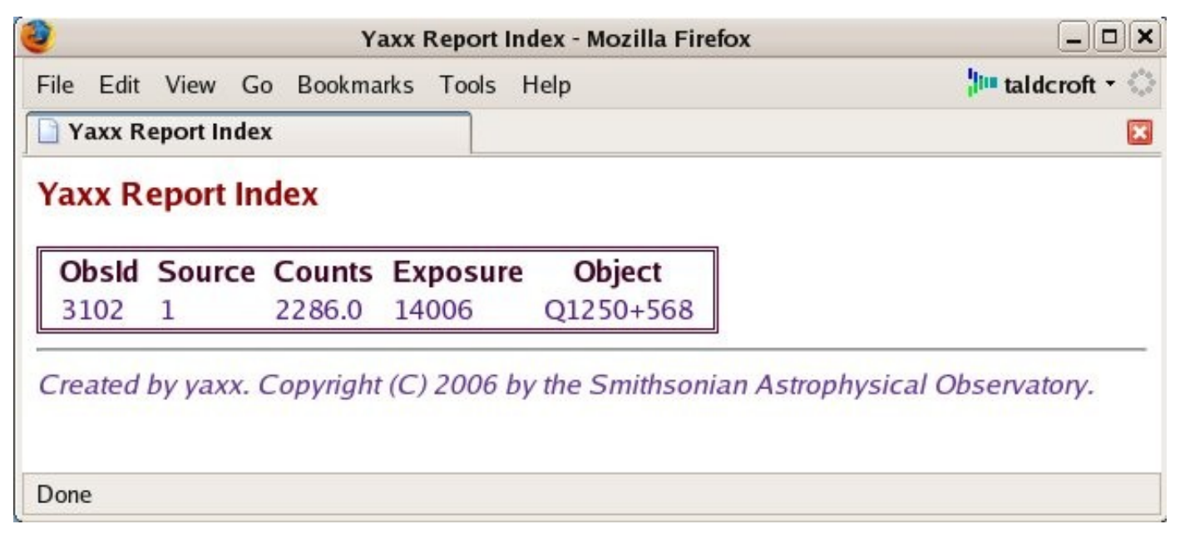
Fit 17 sources with default models

Edit configuration file to set input data directory

```
input_dir = /data/chandra/obs%d # Input data dirs
```

Create object list file with all sources
Run *yaxx*
View fit results

Master report index



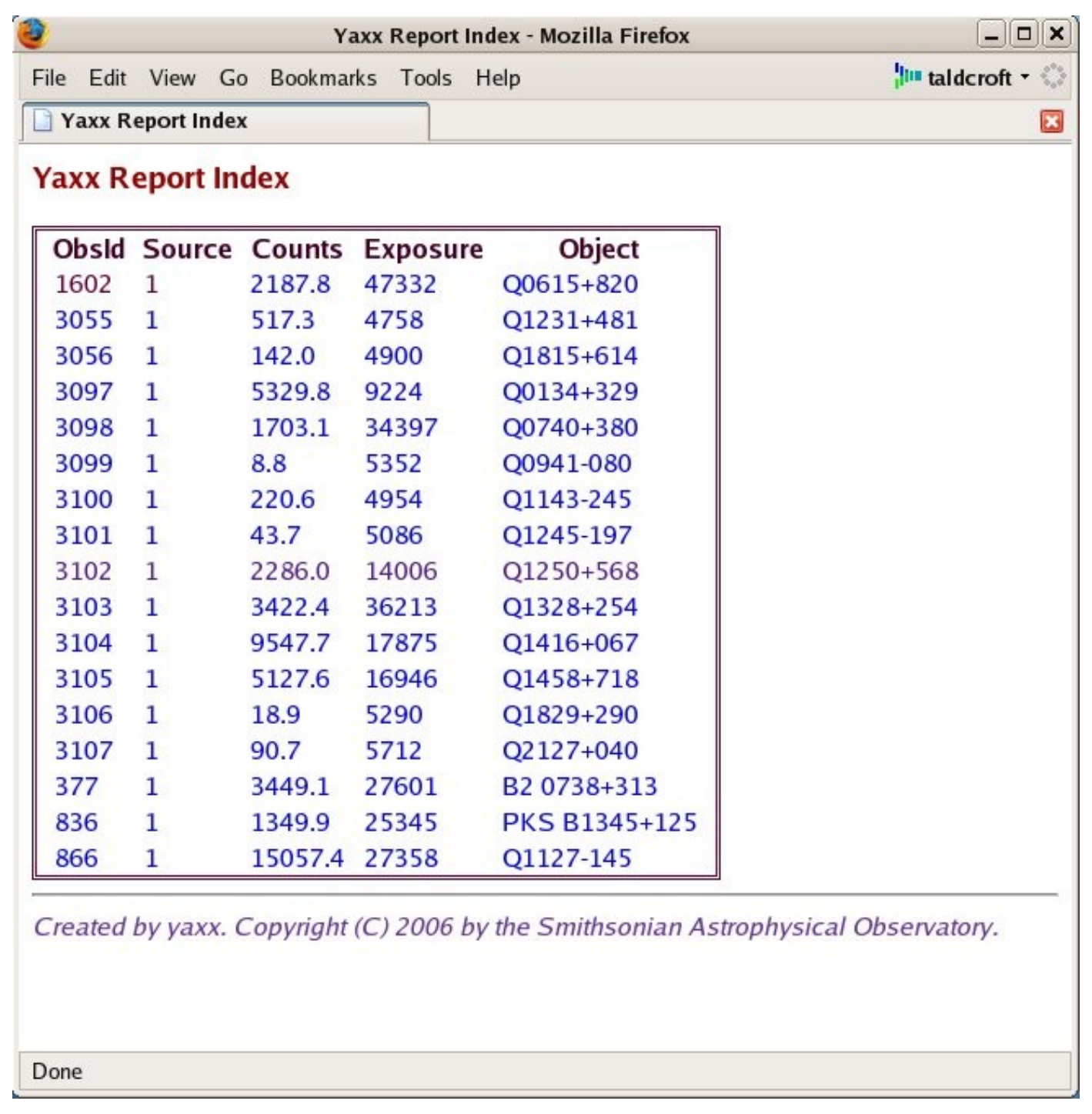
Yaxx Report Index - Mozilla Firefox

Yaxx Report Index

ObsId	Source	Counts	Exposure	Object
3102	1	2286.0	14006	Q1250+568

Created by yaxx. Copyright (C) 2006 by the Smithsonian Astrophysical Observatory.

Done



Yaxx Report Index - Mozilla Firefox

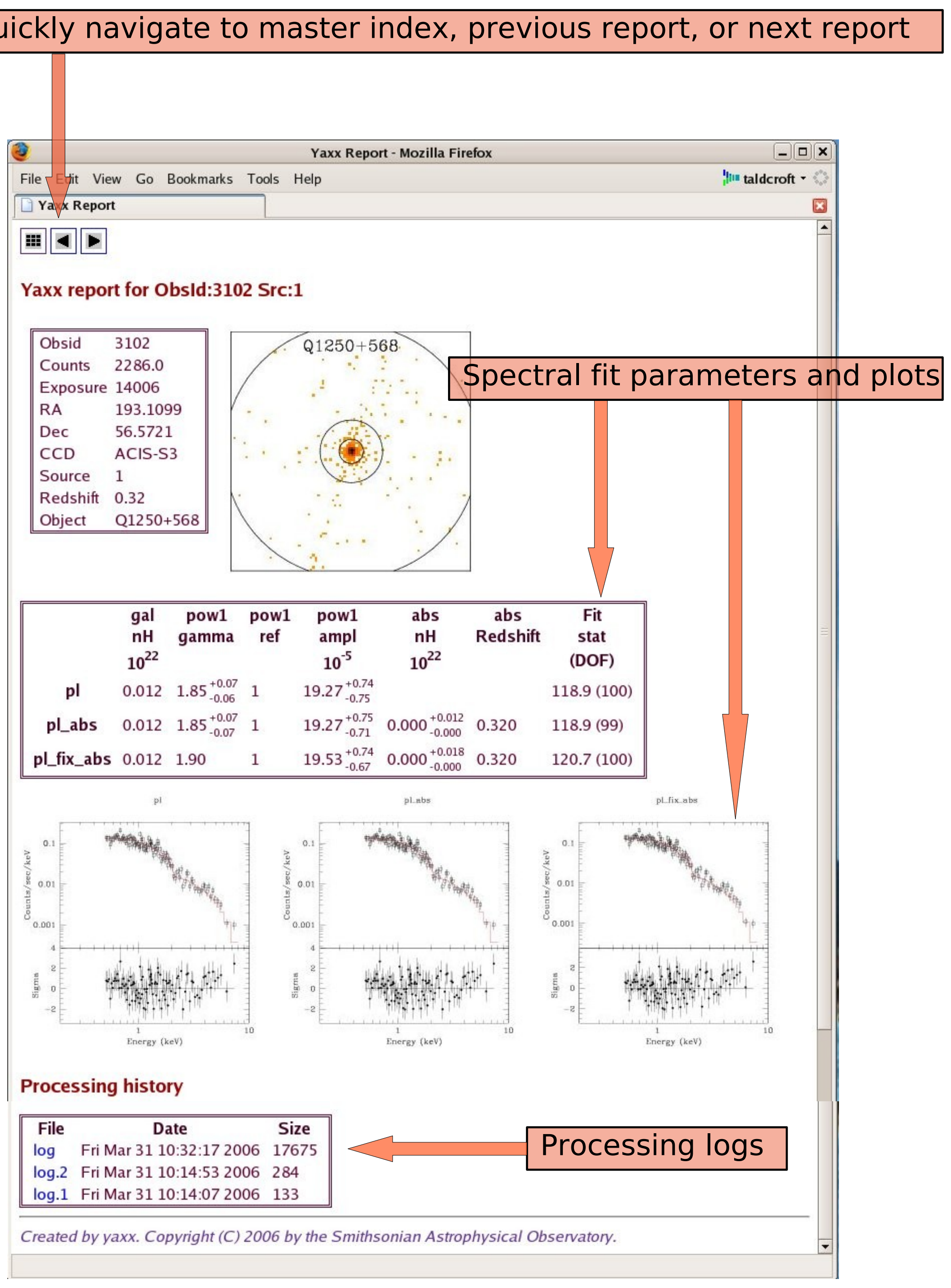
Yaxx Report Index

ObsId	Source	Counts	Exposure	Object
1602	1	2187.8	47332	Q0615+820
3055	1	517.3	4758	Q1231+481
3056	1	142.0	4900	Q1815+614
3097	1	5329.8	9224	Q0134+329
3098	1	1703.1	34397	Q0740+380
3099	1	8.8	5352	Q0941-080
3100	1	220.6	4954	Q1143-245
3101	1	43.7	5086	Q1245-197
3102	1	2286.0	14006	Q1250+568
3103	1	3422.4	36213	Q1328+254
3104	1	9547.7	17875	Q1416+067
3105	1	5127.6	16946	Q1458+718
3106	1	18.9	5290	Q1829+290
3107	1	90.7	5712	Q2127+040
377	1	3449.1	27601	B2 0738+313
836	1	1349.9	25345	PKS B1345+125
866	1	15057.4	27358	Q1127-145

Created by yaxx. Copyright (C) 2006 by the Smithsonian Astrophysical Observatory.

Done

Quickly navigate to master index, previous report, or next report



Yaxx Report - Mozilla Firefox

Yaxx report for ObsId:3102 Src:1

Obsid	3102
Counts	2286.0
Exposure	14006
RA	193.1099
Dec	56.5721
CCD	ACIS-S3
Source	1
Redshift	0.32
Object	Q1250+568

Spectral fit parameters and plots

	gal nH 10^{22}	pow1 gamma	pow1 ref	pow1 ampl 10^{-5}	abs nH 10^{22}	abs Redshift	Fit stat (DOF)
pl	0.012	$1.85^{+0.07}_{-0.06}$	1	$19.27^{+0.74}_{-0.75}$			118.9 (100)
pl_abs	0.012	$1.85^{+0.07}_{-0.07}$	1	$19.27^{+0.75}_{-0.71}$	$0.000^{+0.012}_{-0.000}$	0.320	118.9 (99)
pl_fix_abs	0.012	1.90	1	$19.53^{+0.74}_{-0.67}$	$0.000^{+0.018}_{-0.000}$	0.320	120.7 (100)

Processing history

File	Date	Size
log	Fri Mar 31 10:32:17 2006	17675
log.2	Fri Mar 31 10:14:53 2006	284
log.1	Fri Mar 31 10:14:07 2006	133

Processing logs

Created by yaxx. Copyright (C) 2006 by the Smithsonian Astrophysical Observatory.

Yaxx is not Simple

Processing threads

- Configuration and processing flow required for each type of data analysis is encapsulated in a processing thread
- Yaxx currently has standard threads for *Chandra* and *XMM*
- A derivative of the *Chandra* thread is used for ChaMP processing
- A major survey project could develop a standard thread to provide a unified configuration for outputs, models etc for individual studies within the project
- Detailed data processing is defined entirely within the thread configuration file, making it possible to tweak existing threads or define completely new threads (possibly having nothing to do with spectral fitting!)

```
# Generate GTI table to filter out times of high rates
<process_step>
  name        create_pn_gti_table
  dir          %FILE{obsid_dir}%
  detector     pn
  depend_file  %FILE{obsid_bgd_rate}%
  target_file  %FILE{obsid_gti}%
  command <<COMMAND
    tabgtigen
      table=%FILE{obsid_bgd_rate}%
      expression='RATE<=1.0'
      gtiset=%FILE{obsid_gti}%
      -w 0
COMMAND
</process_step>
```

Target dependencies

%FILE{}% macro lets *yaxx* worry about file names. In this case the name includes the XMM detector.

Command definition

Configuration

- Most behavior of *yaxx* is controlled by a hierarchical set of configuration files, from System level (global options not changed after installation) down to options specific to a single source.
- Spectral fitting models are completely configurable
- Choice of fit model(s) determined by *yaxx* for each source based on user-supplied criteria
- Output report format specified by HTML and LaTeX templates

Example: Calculate unabsorbed luminosity

- Using the powerful combination of the Sherpa, S-lang, and the yaxx macro language, one can do complex manipulations and capture the results in the output data
- The following is placed in the *Sherpa* fit template:

```
# Calculate the 2-8 keV (rest-frame) unabsorbed luminosity
e2_rest = 2.0 / (1+%VALUE{redshift}%)
e8_rest = 8.0 / (1+%VALUE{redshift}%)
unabs_flux2_8_rest = get_eflux(1, [e2_rest, e8_rest], "pow1")
flux_cmd = "flux_to_luminosity.pl -redshift %VALUE{redshift}% -flux "
           + string(unabs_flux2_8_rest.value)
fp = popen(flux_cmd, "r")
lines = fgetslines(fp)
()=pclose(fp)
unabs_lumin_2_8_rest = lines[0]
fits_update_key(fp, “LU20_80R”, unabs_lumin_2_8_rest, “Luminosity”)
```

Grab results

Run an external perl script

Save as MDL file header keyword

Yaxx real-world use and development

- *Yaxx* supplies core data processing functions for three major survey projects: ChaMP, ANCHORS, and C-COSMOS
- Several analysis projects which are **not** concerned with X-ray data now take advantage of the capabilities of *yaxx*
- Substantial development continues, primarily focused on refactoring code to split out core processing functions from specific analysis thread (e.g. X-ray spectral fitting)
- *Yaxx* is no longer Yet Another X-ray Xtractor. *Yaxx* is *yaxx*!

Download

- Further information and source download at **http://cxc.harvard.edu/contrib/yaxx**

Acknowledgements

- *Yaxx* has been developed with support from NASA grant NAS8-03060 and CXC archival research grant AR2-3009X