Yet Another X–ray Xtractor

Tom Aldcroft, SAO/CXC

http://cxc.harvard.edu/contrib/yaxx

Yaxx is a Perl script that facilitates batch spectral processing of X-ray data using Perl open source tools and commonly available astronomical software (CIAO/*Sherpa*, SAS, HEAsoft). 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 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, 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
- Facility to easily do Monte-Carlo simulations
- Completely free, as in beer and only open source software
- Supported on Linux and Solaris platforms, with Mac OS X support in work

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
```

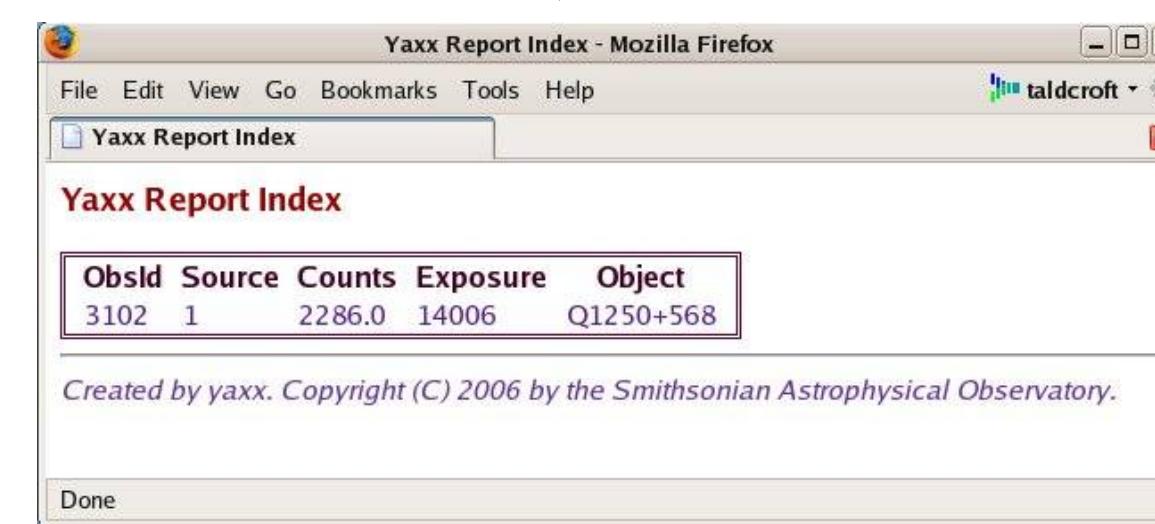### Fit 17 sources with default models

Edit configuration file to set input data directory
```
input_dir = /data/chandra/obs%d # Input data dirs
```

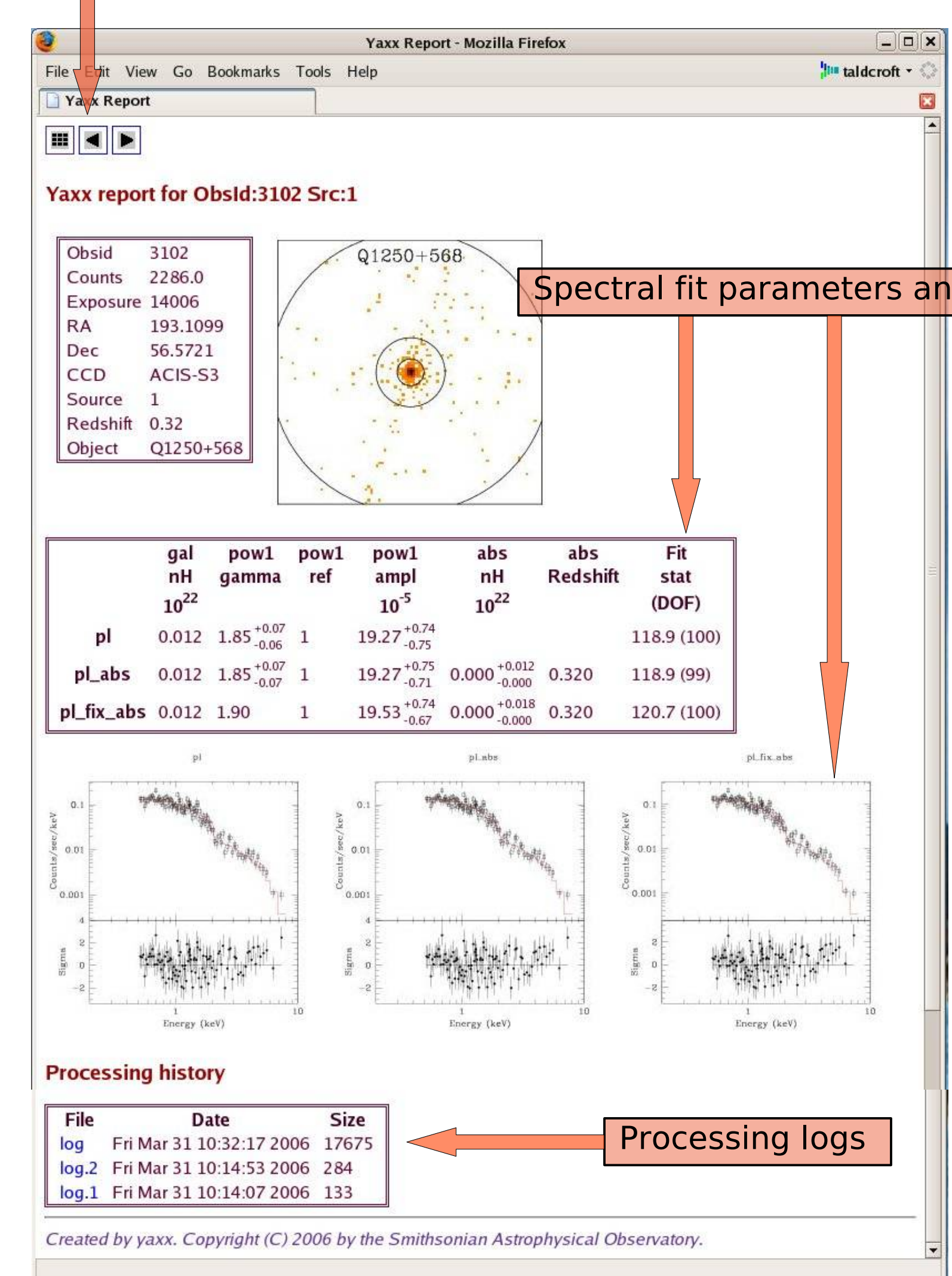Create object list file with all sources
Run *yaxx*
View fit results

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>
```

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], "powl")
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")
```

Real world applications of *yaxx*

- ChaMP (Silverman et al. 2005 ApJ 618 123; Green et al. Extragalactic Surveys poster)
- ANCHORS: An archive of Chandra observations of regions of star formation (http://cxc.harvard.edu/ANCHORS)
- "Evolution of the X-ray Emission of Radio-Quiet Quasars" (Kelly et al. 2006, ApJ Letters accepted)
- "X-ray Properties of the Gigahertz-peaked and Compact Steep Spectrum Sources" (Siemiginowska et al. *in prep*)

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