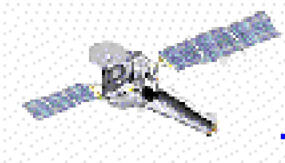


GUIs for All

Why the GUI Way?

- occasional users of *Chandra* data
  - ⇒ do want to get products fast and easily
  - ⇒ do not want to learn CIAO or S-LANG

- many more of such users as archive grows and L3 progresses

- we want to make simple analysis simple (where possible) even for experienced users.

  Example: *extract spectrum, ARF, RMF of source and background with simple `DS9` clicks (~`acisspec`), extract source statistics*

- software survey and helpdesk tickets: large usage of `ds9` (image display) and of `prism` (file browser)

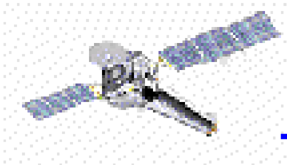
Several Kinds of GUIs

In CIAO 3 we have examples of three classes of GUI:

- Single purpose - `peg` (parameter editor)
- Multi purpose - `ds9, prism`
- "do-it-all" - the `firstlook` concept

- Developing and using `firstlook` taught us that designing a flexible, powerful, easy-to-use data analysis GUI requires a significant amount of resources.

- We now have the infrastructure (both tools and libraries) in place
- Use of `S-Lang` and `SL-GTK` allows us quick prototyping of GUIs
  - ⇒ Example of a prototype "gui-let" (a DIY approach for specialised projects) to configure Sherpa



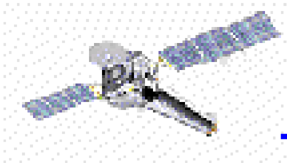
Future Directions

- Expand capabilities of current GUIs:
  - more features to prism (e.g. statistics)
  - more scripts to `ds9, prism`
  - `firstlook`

- GUI interface(s) for `SHERPA` (more user-friendly for the X-ray and *non* X-ray community)

- GUI interface for `CHIPS` (better/easier visualization wanted by many CIAO users in survey)

- Create single or multi-purpose GUIs for specific CIAO tasks, such as a thread editor or Chart configurator



A “GUI-let” Example: The Sherpa Configuration GUI

- In CIAO 3, *Sherpa* can be configured by use of a S-Lang structure
  - ⇒ good for automated analysis
  - ⇒ hard to remember for the casual user
  - ⇒ simpler with a GUI (written in S-Lang/SL-GTK)

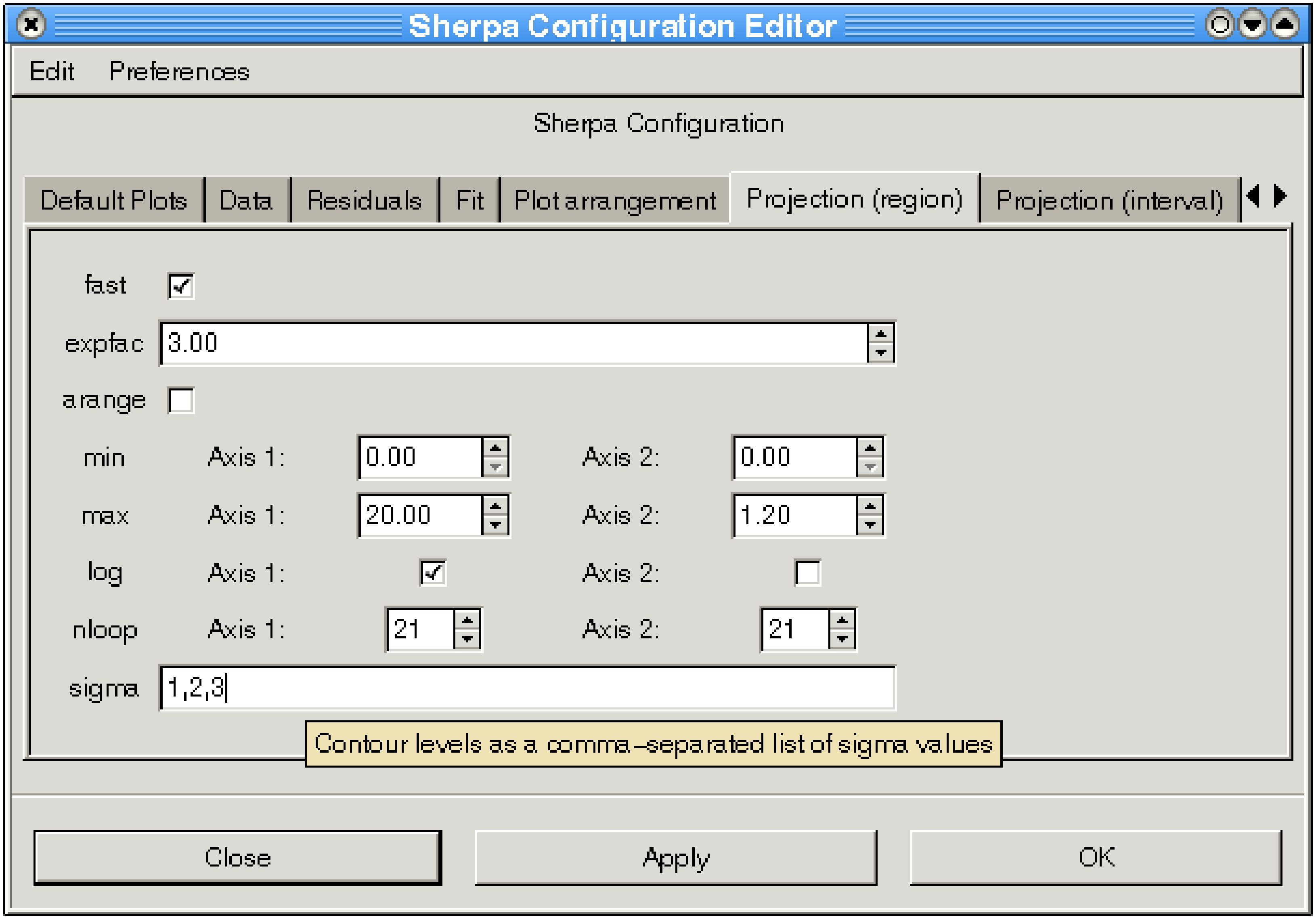
S-Lang/Sherpa

```
sherpa> sherpa.regproj.arange = 0
sherpa> sherpa.regproj.min = [ 1, 0 ]
sherpa> sherpa.regproj.max = [ 20, 1.2 ]
sherpa> sherpa.regproj.log = [ 1, 0 ]
sherpa> sherpa.regproj.nloop = [ 21, 21 ]
sherpa> reg-proj clus.kt clus.abundanc
```

versus GUI

```
sherpa> sherpa_config ....clicks on GUI button ...
sherpa> reg-proj clus.kt clus.abundanc
```

Prototype


Sherpa Configuration Editor
Edit
Preferences
Sherpa Configuration
Default Plots
Data
Residuals
Fit
Plot arrangement
Projection (region)
Projection (interval)
fast
expfac 3.00
arange
min Axis 1: 0.00 Axis 2: 0.00
max Axis 1: 20.00 Axis 2: 1.20
log Axis 1: Axis 2:
nloop Axis 1: 21 Axis 2: 21
sigma 1,2,3
Contour levels as a comma-separated list of sigma values
Close
Apply
OK

The results of the projection are shown below