X-RAY SPECTROSCOPY:
PART I - DATA BASICS

MICHAEL A. NOWAK
(MIT-CHANDRA X-RAY SCIENCE CENTER)

-WITH HELP OVER THE YEARS FROM-
JÖRN WILMS, JOHN DAVIS, JOHN HOUCK,
DAVE HEUNEMOERDER, MIKE NOBLE
OUTLINE (PART 1):

☐ THE COMPONENTS OF AN X-RAY SPECTRAL FIT:
  ☐ DATA, SPECTRUM, RMF, ARF, BACKGROUND
  ☐ LOADING & VISUALIZING THE DATA
  ☐ DEFINING A MODEL
  ☐ ISIS EXAMPLES SPRINKLED THROUGHOUT
WHAT YOU WANT TO UNCOVER: ASTROPHYSICAL SPECTRA

$\bar{I}_\nu(\vec{x})$

DETECTOR

IGM, ISM, ETC.

SOURCE

BACKGROUND
INTENSITY AS A FUNCTION OF POSITION & DIRECTION

- BUT, LIMITATIONS ABOUND...

- LACK OF SPATIAL RESOLUTION - SPATIAL RESOLUTION:
  - CHANDRA -> XMM -> SUZAKU -> RXTE

- FOREGROUND ABSORPTION, BACKGROUND EMISSION

- LACK OF ENERGY RESOLUTION - ENERGY RESOLUTION:
  - CHANDRA (HETG) -> XMM (RGS) -> SUZAKU -> RXTE

- LIMITED COLLECTING AREA -> COUNTS, NOT FLUX!
X-RAY SPECTROSCOPY IS ABOUT COUNTS PER CHANNEL

- $\tilde{I}_\nu(x)$ is: ENERGY/SEC/Hz/AREA/STERADIAN

- X-RAY TELESCOPES MEASURE:
  COUNTS/(INTEGRATED TIME BIN)/"CHANNEL" - TEMPORAL RESOLUTION:

- RXTE -> XMM -> CHANDRA -> SUZAKU

- COUNTS ARE NOT PHOTONS!

- POISSON STATISTICS ARE COMMONLY USED

- CHANNELS ARE NOT ENERGY!
WE CONVERT FROM FLUX* TO DETECTOR COUNTS§

\[ C(h) = \int_0^\infty \sum_i R_i(h, E) A_i(E) S_i(E) \, dE \, dT + B(h) \]

DETECTED COUNTS
RESPONSE MATRIX
EFFECTIVE AREA
SOURCE SPECTRA
BACKGROUND COUNTS

CAN “OVERLAP”

(SEE J. DAVIS, 2001, APJ, 548, P. 1010)

*NOT JUST FROM THE SOURCE
§ALSO NOT JUST FROM THE SOURCE
WHAT ARE THESE PIECES?

- $S(E)$: SPECTRAL ENERGY DISTRIBUTION, UNITS = PHOTONS/SEC/AREA/ENERGY

- $A(E)$: EFFECTIVE AREA/ANCILLARY RESPONSE FUNCTION/ARF, UNITS = AREA/PHOTON

- $R(h, E)$: RESPONSE FUNCTION/RMF, UNITLESS & SOMETIMES NORMALIZED:
  $$\sum_h \int_0^E R(h, E) \delta(E - E_0) dE = 1$$

- $dE, dT$: PHOTON ENERGY, INTEGRATION TIME

- $C(h), B(h)$: SOURCE & BACKGROUND COUNTS (EVENTS)

- $h$: PULSE HEIGHT ANALYSIS (PHA) OR PULSE INVARIANT (PI) CHANNEL. DISCRETE!!!
EFFECTIVE AREA (ARF):

ARF CAN HAVE MANY COMPONENTS

- DETECTORS
- EXTRATION REGION DEPENDENT
- FILTERS, WINDOWS, SHIELDS
- MIRRORS, COLLIMATORS
- CAN BE ANGLE DEPENDENT
EFFECTIVE AREA (ARF):

- ARF - combinations of telescope collecting area, detector efficiencies, filter throughputs, etc.

- "Structure" often related to physics of the mirrors/collimators, detectors, filters, etc.

- Typically highest "on-axis", lowest "off-axis". "Off-axis" can be up to ~1° (RXTE).

- Unresolved/off-axis sources = background

- Depends upon "extraction region". Get the whole point spread function (PSF)? Extract only outer radii (e.g., Suzaku)? Detector moving? etc.
EFFECTIVE AREA (ARF):

- PSF OUTSIDE BIG CIRCLE WAS EXCLUDED
- PSF INSIDE SMALL CIRCLE WAS EXCLUDED (PILE UP)
- SUZAKU CREATES ARFS VIA MONTE CARLO SIMULATIONS (SPACECRAFT SPECIFIC TOOL!)
EFFECTIVE AREA (ARF):

- ARF created by satellite-specific software!
- Can be visualized (see exercise):

```plaintext
isis> arf = load_arf("/path/to/arf/file.arf");
isis> plot_bin_integral;
isis> % Area vs. wavelength (Angstrom):
isis> hplot(arf.bin_lo,arf.bin_hi,arf.value);
isis> % Area vs. energy (keV):
isis> hplot(_A(arf.bin_hi),_A(arf.bin_lo),reverse(arf.value));
```
EFFECTIVE AREA (ARF):

SOME OF THESE FEATURES ARE “FIXED”, OTHERS CAN VARY FROM OBSERVATION TO OBSERVATION
RESPONSE FUNCTION (RMF):

- Photon has made it past filters, mirrors, detectors, how is it actually registered?
- "Pulse height" - charge cloud, current, etc.
- Energy/flux -> amplitude/peaks or counts
- Input energy -> output channel is one -> many
- RMF encodes information about detector resolution, and other physical effects
- Resolution can follow its own "Poisson statistics", and often goes as $E^{1/2}$
- Gratings are spatial, follow $\Delta \Delta \sim$ constant
RESPONSE FUNCTION (RMF):

- RMF CREATED BY SATELLITE-SPECIFIC SOFTWARE!

- CAN BE VISUALIZED (SEE EXERCISE):

```matlab
isis> load_rmf("/path/to/rmf/file.rmf");
isis> assign_rmf (1,1);
isis> fit_fun("delta(1)"hift);
isis> set_par (1,1);
isis> set_par (2,_A(6.0)); % 6 keV
isis> eval_counts;
isis> plot_unit("a"); ylog; plot_model(1); % Angstrom
isis> plot_unit("keV"); plot_model(1); % keV
```

- THIS IS JUST FOLDING A DELTA-FUNCTION THROUGH THE COUNTS EQUATION, YIELDING $C(h) = \text{RMF}(h,E_0)$
RESPONSE FUNCTION (RMF):

CHANDRA-ACIS RESPONSE

RMF @ 6 keV, 2.0664 A, sum=0.99983, moment=5.95506
RESPONSE FUNCTION (RMF):

RXTE - PCA RESPONSE

(COURTESY J. DAVIS)
RESPONSE FUNCTION (RMF):

- Fluorescence peaks are caused by the same processes as in astrophysical sources.
- Photon with energy > detector material edge energy (xenon gas, silicon wafer) knocks out inner shell electron.
- L -> K transition yields fluorescence photon, which might be detected.
- Low energy transitions yield escape photon(s) which might be detected.

\[ E_{\text{escape}} = E_{\text{incident}} - E_{\text{fluorescence}} \]
RESPONSE FUNCTION (RMF):

CYG X-1 VIEWED WITH CHANDRA-HETG

CAN COMPARE “CCD ENERGY” WITH “GRATINGS ENERGY”
RESPONSE FUNCTION (RMF):

\[ |m\lambda| \frac{E}{hc} = m\lambda = \text{TG}_\text{MLAM} \]

\[ E = \text{ENERGY} \]

Escape Events

Fluorescence Events

RMF profiles

(COURTESY J. DAVIS)
RESPONSE FUNCTION (RMF):

CYG X-1: FROM HTTP://TGCAT.MIT.EDU

CHANGL

CCD ENERGY

MEG all

GRATINGS ENERGY

order*Wavelength [ Angstroms ]

CHANNEL

CCD ENERGY
RESPONSE FUNCTION (RMF):

- Labeling channels with an energy is a convenience!
- Energy is continuous, PHA bins are discrete.
- The mapping of “PHA” values to energy is the gain.
- This mapping can vary from position to position on the detector. In principle, this is not a problem since the ARF & RMF should encode the mappings.
- Sometimes it's convenient to have the same energies correspond to the same “PHA” values.
- “PHA” -> “PI” = “PULSE INVARIANT”
RESPONSE FUNCTION (RMF):

\[
C(h) = \int_0^\infty \sum_i R_i(h, E) A_i(E) S_i(E) \, dE \, dT + B(h)
\]

- Gain mapping (PHA ↔ Energy) is not always correct. Can add a correction in the fit process.
- If fit disagrees with "known" line energy.
- Sometimes for convenience, RMF & ARF are combined in a single response ("resp") file.
- But not always, especially when some parts change with time, while others do not.
BACKGROUND

\[ C(h) = \int_0^\infty \sum_i R_i(h, E) A_i(E) S_i(E) \, dE \, dT + B(h) \]

- **UNRESOLVED SOURCES** CAN BE BACKGROUND. “X-RAY BACKGROUND”, GALACTIC RIDGE EMISSION, ETC.

- “RESOLVED”, BUT DIFFUSE EMISSION CAN BE BACKGROUND. DUST, HOT GAS, ETC.

- IN PRINCIPLE, THESE COULD (SHOULD) BE ADDED TO S(E)

- BACKGROUND CAN BE INSTRUMENTAL (DETECTOR NOISE), NON X-RAY (COSMIC RAYS), PARTICLE BACKGROUND, ETC. THESE TYPES OF BACKGROUNDS ARE ADDED TO B(h).
BACKGROUND

\[ C(h) = \int_0^{\infty} \sum_i R_i(h, E) A_i(E) S_i(E) \, dE \, dT + B(h) \]

☐ BACKGROUND MIGHT BE MODELED OR MEASURED

☐ MODELED: ADDED TO THE S(E) -OR- B(h) TERMS. I.E., MAY, OR MAY NOT, BE “FOLDED THROUGH” RESPONSE

☐ IF MODELED WITH FIT PARAMETERS, (USUALLY) NO CHANGES TO THE STATISTICS.

☐ MEASURED: USUALLY ADDED TO THE B(h) TERM. NO “FOLDING THROUGH” THE RESPONSE.

☐ (USUALLY) CHANGES THE STATISTICS DEFINITION.
BACKGROUND

\[ C(h) = \int_0^\infty \sum_i R_i(h, E) A_i(E) S_i(E) \, dE \, dT + \boxed{B(h)} \]

□ ISIS NEVER “SUBTRACTS” THE BACKGROUND. IT IS ADDED TO “MODEL” TERMS, AND COMPARED TO *TOTAL COUNTS*.

□ THE ONLY QUESTION IS, DO YOU REDEFINE THE COMPARISON STATISTICS?

□ DEFAULT STATISTICS (CHI² BASED UPON DATA COUNTS) DOES CHANGE WITH BACKGROUND.

□ THIS DEFAULT BEHAVIOR CAN BE ALTERED.
BACKGROUND

Source background = Measured background * (source area) * (source time) / [(background area) * (background time)]

\[
B_s(h) = B_m(h) \times \frac{A_s(h) T_s(h)}{A_m(h) T_m(h)}
\]

This information is usually stored in the spectrum data files & background data files via the “exposure” and “backscale” keywords. (usually no h dependence of a & t).

Defaults incorporate “background model” uncertainty in the comparison statistic:

\[
\Delta C(h) = \sqrt{C(h) + \left( \frac{A_s(h) T_s(h)}{A_m(h) T_m(h)} \right)^2 B_m(h)}
\]

\[
= \sqrt{C(h) + \left( \frac{A_s(h) T_s(h)}{A_m(h) T_m(h)} \right) B_s(h)}
\]

-Or- from non-simultaneous measurements, “blank sky” observations, detector models, etc.
BACKGROUND

- Backgrounds often created by satellite-specific software!
- Can be accessed and visualized (see exercise):

```isidiff
% Load the data, including background if in header
isis> d = load_data("/path/to/obs/data.pha");
isis> ch = get_data_counts(d); % Data & bin edges
isis> bs = get_back(d); % "Scaled" background
isis> ts = get_data_exposure(d); % T_s(h)
isis> as = get_data_backscale(d); % A_s(h)
isis> tm = get_back_exposure(d); % T_m(h)
isis> am = get_back_backscale(d); % A_m(h)
isis> plot_bin_integral;
isis> hplot(_A(ch.bin_lo),_A(ch.bin_hi), reverse(ch.value)); % keV plot
isis> ohplot(_A(ch.bin_lo),_A(ch.bin_hi), reverse(bs)); % keV plot
```
SOURCE MODEL

\[ C(h) = \int_0^\infty \sum_i R_i(h, E) A_i(E) S_i(E) dE \, dT + B(h) \]

- Saved for last, since in some ways this is the least important thing you need to worry about, but what most people only want to think about...

- Models are computed on a discrete grid, *finer* than the \( h \)-grid. (Sometimes ridiculously so.)

- Outputs of these models are: counts/sec/cm\(^2\)/*bin*

- The content of these models is astrophysics... i.e., the subject of the next 3-5 years of your life
ISIS DEFAULTS TO WAVELENGTH GRID, WITH LO & HI BINS EXPLICITLY SPECIFIED. (IT HAS WRAPPERS TO INTERFACE WITH EXISTING XSPEC MODELS.) MODELS ARE COUNTS INTEGRATED OVER THE BIN WIDTHS!

ISIS _A() FUNCTION CONVERTS ANGSTROMS <-> KEV

KEEP THIS IN MIND WHEN ADDING NEW FUNCTIONS!

isis> define qpo_fit(lo,hi,par){
    variable l,rms,qpo,q,f,al,ah;

    % Go from Angstrom to keV (which we pretend is Fourier Hz),
    al = _A(lo); ah = _A(hi);
    rms = par[0]; q = par[1]; f = par[2]; % RMS, Q-value, Frequency
    qpo = rms/(0.5 - atan(-2.*q)/PI);
    qpo = qpo^2/(al-ah)/PI*( atan(2.*q*(al-f)/f) - atan(2.*q*(ah-f)/f) );
    return = reverse(qpo);
}
isis> add_slang_function("qpo",["norm [rms]","Q [f/FWHM]","f [Hz]"]);
THE PIECES WE NEED:

- **FITS (FLEXIBLE IMAGE TRANSPORT SYSTEM) - BINARY FORMAT TO STORE ASTROPHYSICAL DATA**
  

- **FILES READ WITH FTOOLS/CFITSIO (HEASOFT), DMTOOLS (CIAO), OTHERS. INCORPORATED INTO ANALYSIS SYSTEMS.**

- **PHA (SPECTRAL FILES) HAVE TYPE 1 & TYPE 2.**

- **TYPE 1 - SINGLE SPECTRUM PER FILE. “HEADER” (WHAT GRPPHA ALTERS) CAN CONTAIN INFORMATION ON ASSOCIATED RESPONSES AND BACKGROUND.**

- **TYPE 2 - MULTIPLE SPECTRA PER FILE (E.G., MULTIPLE GRATING ORDERS), NO STANDARD STORAGE OF NAMES OF ASSOCIATED RESPONSES AND BACKGROUND.**
THE PIECES WE NEED:

- SATELLITE SPECIFIC PIECES: PHA, RMF, ARF, BKG = FITS FILES. CAN BE ASSOCIATED VIA FTOOL FUNCTION “GRPPHA”.

UNIX%> GRPPHA HXTA.PHA
** GRPPHA 3.0.1
....... USING PHA_GP VER 1.1.1
PLEASE ENTER OUTPUT FILENAME[] !HXTA.PHA

---------------------------------------------------
MANDATORY KEYWORDS/VALUES
---------------------------------------------------

- EXTNAME - SPECTRUM NAME OF THIS BINTABLE
- TELESCOP - XTE MISSION/SATELLITE NAME
- INSTRUME - HEXTE INSTRUMENT/DETECTOR
- FILTER - NONE INSTRUMENT FILTER IN USE
- EXPOSURE - 7086.5 INTEGRATION TIME (IN SECS) OF PHA DATA
- AREASCAL - 1.0000 AREA SCALING FACTOR
- BACKSCAL - 1.0000 BACKGROUND SCALING FACTOR
- BACKFILE - HXTABACK.PHA ASSOCIATED BACKGROUND FILE
- CORRSCLAL - 0.0000 CORRELATION SCALING FACTOR
- CORRFILE - NONE ASSOCIATED CORRELATION FILE
- RESPFILE - HEXTE-A_SRC.RMF
- ANCRFILE - HEXTE-A_SRC.ARF
- POISSERR - FALSE WHETHER POISSONIAN ERRORS APPLY
- CHANTYPE - PHA WHETHER CHANNELS HAVE BEEN CORRECTED
- TLMIN1 - 0 FIRST LEGAL DETECTOR CHANNEL
- DETCHANS - 256 NO. OF LEGAL DETECTOR CHANNELS
- NCHAN - 256 NO. OF DETECTOR CHANNELS IN DATASET
- PHAVERSN - 1.2.0 OGIP FITS VERSION NUMBER
- STAT_ERR - TRUE STATISTICAL ERROR
- SYS_ERR - FALSE FRACTIONAL SYSTEMATIC ERROR
- QUALITY - TRUE QUALITY FLAG
- GROUPING - FALSE GROUPING FLAG

---------------------------------------------------

GRPPHA CAN ALSO BE USED TO:

- BIN DATA
- SET SYSTEMATIC ERROR BARS
- SET CORRFILE (ADDITIONAL BACKGROUND)
- WE WILL DO ALL OF THESE DURING ISIS ANALYSIS.
X-RAY SPECTROSCOPY ANALYSIS PACKAGES

☑ Packages are designed to read & plot the data.

☑ Handle the “meta-data”, i.e., proper association of exposures, responses, & backgrounds.

☑ Keep track of “grouping” & noticed/ignored bins.

☑ Minimize the fit statistic(s), and report the values of “best-fit” parameters and statistics.

☑ Calculate parameter error bars, fluxes, etc.

☑ Save and plot results.

☑ Today

☑ Monday & Tuesday
COMMON ANALYSIS PACKAGES:

- There are many analysis packages out there, ranging from custom IDL codes to well-established, long used programs (XSPEC).

- XSPEC is the most commonly used, with a wide range of intrinsic & user contributed (“local”) models: http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec

- A variety of other analysis packages are used:
  - ISIS: http://space.mit.edu/CXC/ISIS/
  - SHERPA: http://cxc.harvard.edu/sherpa/
  - SPEX: http://www.sron.nl/divisions/hea/spex/
COMPARISON OF SOME ANALYSIS PACKAGE FEATURES:

<table>
<thead>
<tr>
<th></th>
<th>XSPEC MODELS</th>
<th>XSPEC LOCAL MODELS</th>
<th>SCRIPTED MODELS</th>
<th>USER SCRIPTS</th>
<th>DATA PRODUCT ACCESS</th>
<th>OTHER FIT KERNEL</th>
<th>USER FIT KERNEL</th>
<th>USER OPTIM. METHS.</th>
<th>USER FIT STATS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ISIS</strong></td>
<td>Nearly All</td>
<td>Yes</td>
<td>S-lang</td>
<td>S-lang</td>
<td>Yes</td>
<td>Gain Pileup</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Sherpa</strong></td>
<td>Most</td>
<td>With Effort</td>
<td>Python</td>
<td>Python</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>XSPEC</strong></td>
<td>All</td>
<td>Yes</td>
<td>Limited-mdefine</td>
<td>TCL</td>
<td>Very Limited</td>
<td>Gain</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>SPEX</strong></td>
<td>Few</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NON-X-RAY DATA</th>
<th>ATOMIC DATA ACCESS</th>
<th>MULTI-CORE ERRORS</th>
<th>MULTI-CORE FITS</th>
<th>MULTI-SYSTEM ERRORS</th>
<th>MULTI-SYSTEM MODELS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ISIS</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Sherpa</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>XSPEC</strong></td>
<td>With Fake RMF,ARF</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>SPEX</strong></td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
INTERACTIVE SPECTRAL ANALYSIS SYSTEM (ISIS)

- SCRIBTABLE & WIDE SELECTION OF USEFUL MODULES:
  - GNU SCIENTIFIC LIBRARY (GSL),
  - STATISTICS & HISTOGRAMS
  - HDF5 (BINARY DATA FORMAT - USED IN ASTRO/HYDRO SIMULATIONS)
  - XPA (INTERFACE TO DS9 ASTRO IMAGER)
  - GTK (GUI'S) & VOLVIEW (3D VISUALIZATION)
  - PARALLEL VIRTUAL MACHINE
  - EXTENSIVE HISTORY OF INCORPORATING PARALLELIZATION METHODS
USING ISIS TO ANALYZE SIMPLEX X-RAY SPECTRA

☐ IN WHAT FOLLOWS, I WILL BE ASSUMING WE'VE LOADED UP THE ISIS UTILITY FUNCTIONS PROVIDED AT:
http://space.mit.edu/home/mnowak/isis_vs_xspec/download.html

☐ ISIS COMMANDS UNIQUE TO THESE SCRIPTS WILL BE HIGHLIGHTED IN RED

☐ NOTE THAT ISIS ALLOWS YOU TO CHANGE THE NAMES OF ANY COMMANDS. DON'T LIKE A NAME? CHANGE IT!

isis> alias(“fit_fun”, “model”);
STARTING UP ISIS:

ISIS CAN MAKE USE OF VARIOUS ENVIRONMENT VARIABLES UPON START UP. USEFUL TO PLACE IN, E.G., A ~/.CShRC FILE.

# If GNU readline has been set, saves input history. VERY USEFUL!!!
setenv ISIS_HISTORY_FILE /home/mnowak/.isis_history

# Places to search for S-lang scripts
setenv SLANG_LOAD_PATH /home/mnowak/slang_scripts
setenv ISIS_LOAD_PATH /home/mnowak/slang_scripts

# Places to search for modules. Modules are compiled binaries with
# S-lang interfaces, e.g., GSL or histogram modules
setenv SLANG_MODULE_PATH /home/mnowak/slang_modules
setenv ISIS_MODULE_PATH /home/mnowak/slang_modules

# The editor we will use when editing model parameter files
# You might prefer to use vi instead.
setenv EDITOR /usr/bin/emacs
# STARTING UP ISIS:

Upon start up, ISIS loads the contents of `~/.isisrc` - file containing S-lang commands & functions.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isis_Append_Semicolon=1;</td>
<td>No ';'; required interactively</td>
</tr>
<tr>
<td>Isis_List_Filenames=1;</td>
<td>Be verbose with list functions</td>
</tr>
<tr>
<td>Isis_Use_PHA_Grouping=1;</td>
<td>Use the grppha applied grouping</td>
</tr>
<tr>
<td>Fit_Verbose=0;</td>
<td>Only final statistics printed</td>
</tr>
<tr>
<td>Isis_Verbose=0;</td>
<td>Not too verbose on ISIS messages</td>
</tr>
<tr>
<td>Minimum_Stat_Err=1.e-30;</td>
<td>=0 forces Poisson errors; we choose a value for when we fit radio/IR data</td>
</tr>
<tr>
<td>Rmf_OGIP_Compliance=2;</td>
<td>Lower number to read poorly written RMFs/ARFs (INTEGRAL requires even more)</td>
</tr>
</tbody>
</table>

```
putenv("PGPLOT_BACKGROUND=white"); % invert default for nicer
putenv("PGPLOT_FOREGROUND=black"); % look - black on white

% Define places to search for useful code.
static variable path="/home/mnowak/";
add_to_isis_load_path(path+"isis_code");
add_to_isis_module_path(path+"isis_code");

% You can load-up useful bits of code automatically ...
()=evalfile(path+"isis_code/isis_utility_functions_prerelease_1.5.sl");
()=evalfile(path+"isis_code/isis_utility_functions.sl");
```
OBTAINING HELP IN ISIS:

- The `who;` command will give you a list of variables that you have defined *on the command line*.
- The `.apropos` command will give you a list of globally defined functions (including from scripts!) that contain a given substring.
- The `.help` command will give you a help file for most ISIS intrinsic functions.

```plaintext
isis> a = [0:10]; b = "pizza";
isis> who;
a: Integer_Type[11]
b: pizza
isis> .apropos rmf
Found 12 matches in namespace Global:
all_rmfs                        assign_rmf                        delete_rmf
find_rmf_peaks                  get_rmf_arf_grid                  get_rmf_data_grid
get_rmf_info                    list_rmf                           load_rmf
rebin_rmf                       set_rmf_info                      unassign_rmf
isis> .help load_arf
load_arf

SYNOPSIS
Load an effective area (ARF) file
...
BRIEF WORDS ABOUT S-LANG:

- S-LANG IS A SCRIPTING LANGUAGE WITH STRONG NUMERICAL ABILITIES. SEE: WWW.S-LANG.ORG.
- ALL S-LANG FUNCTIONS WORK IN ISIS, *ALL* ISIS COMMANDS CAN BE USED IN ISIS/S-LANG SCRIPTS.

isis> a=[0:10]; b = sin(2*PI*a/10)+1; % Vector math
isis> variable y = struct{time, rate}; % Some ISIS commands return structures
isis> y.time=[0:100:0.1]; % Note we are using % for comments
isis> y.rate=y.time^2; % Scripts require ; and 'variable' -
isis> % latter not required in ISIS, former can be turned off
isis> ()=load_data(“pca.pha”); % Note that ()= captures and discards
isis> % the return value, otherwise
isis> (a,b,c) = some_funct(b); % Lets say this returns 1,2,3 ...
isis> print(b);
 2
isis> (,,) = some_funct(b); % Discard the return values
isis> some_funct(b);
 3
 2
 1
isis> % Top of the "stack", 3, pops off first, then 2, then 1 ...
ISIS WILL ASSIGN NUMERIC IDENTIFIERS TO DATA SETS, RMFS, & ARFS.

NOTE THAT FOR A GIVEN DATA SET, THESE 3 NUMBERS ARE *NOT* NECESSARILY THE SAME!

LOAD_RMF("RMF.FITS"); LOAD_ARF("ARF.FITS"); WILL LOAD JUST THE RMF/ARF - USED WITH TYPE 2 PHA FILES.

```python
isis> ()=evalfile("/path/./isisrc"); % Auto-loaded if /path=HOME
isis>
isis> pca = load_data("pca_cygx1_I.pha");
RMF includes the effective area
Warning: negative EBOUNDS value E_MIN=-0.161702, set to zero
Warning: 1 hi/lo grid values needed tweaking
isis> print(pca);
1
```
LOADING & PLOTTING DATA:

- FOR RXTE-PCA SPECTRA, RMF & ARF ARE COMBINED. DATA=1, RMF=1, ARF HAS NO NUMBER.
- THE NEXT RMF WE LOAD WOULD BE ASSIGNED #2, THE NEXT ARF WE LOAD WOULD BE ASSIGNED #1

```plaintext
isis> list_data;
Current Spectrum List:
  id   instrument   m  prt  src   use/nbins  A  R  totcts  exp(ksec)
target
  1        PCA  0  0  0   129/  129   -  1  7.8833e+06     9.200
file:  pca_cygx1_I.pha
back:  pcaback_cygx1_I.pha
isis>
isis> list_rmf;
Current RMF List:
  id  grating detector  m  type   file
  1    PCU2    0 file:  pcaresp_cygx1_I.rmf
isis>
isis> list_arf;
isis>
```
LOADING & PLOTTING DATA:

- **INTEGRISTIC ISIS PLOTS ALWAYS LEAVE BACKGROUND IN, AND AREN’T GREAT FOR MULTIPLE DATASETS.**

- **THE CUSTOM ROUTINES (.isisrc_plots) ADD FUNCTIONALITY**

```plaintext
isis> plot_unit("kev");
isis> plot_bin_integral; plot_data_counts(pca);
isis> xlog; ylog; plot_bin_integral; plot_data_counts(pca);
isis> plot_bin_density; plot_data_counts(pca);
isis>
isis> fancy_plot_Unit("kev");
isis>
isis> plot_counts(pca;dsym=4,dcol=4);
isis> plot_data(pca;dsym=4,dcol=4);
isis>
isis> plot_counts(pca;dsym=4,dcol=4,bkg=1);
isis> plot_data(pca;dsym=4,dcol=4,bkg=1);
```
HOW SHOULD I PLOT DATA?

USEFUL *BEFORE* BINNING, *OR* WITH UNIFORM BINS

USEFUL AFTER BINNING

DETECTOR OR BIN FEATURES
PLOTTING WITH BACKGROUND IS USEFUL TO SEE ITS FEATURES
PUTTING BOTH TOGETHER:

isis> plot_data({pca,pca};dsym={4,4},
  dcol={8,4},bkg={1,0});
GROUPING & LIMITING DATA:

- Where to ignore the data is a matter of knowledge of the spacecraft, and experience with its data.
- PCA responses not so good < 3 keV, background starts becoming important above ~20 keV.
- These decisions can change over time, as calibrations improve.
GROUPING & NOTICING

- GROUPING TRADITIONALLY USED TO ACHIEVE SUFFICIENT SIGNAL-TO-NOISE TO USE ~GAUSSIAN STATISTICS WITH ERROR ~ \( (\text{COUNTS})^{1/2} \).

- OFTEN: MINIMUM COUNTS/BIN, UNIFORM CHANNELS/BIN

- IGNORE BINS WITH LOW S/N AND/OR POOR CALIBRATION

- GRPPHA (HEASOFT) OR DMGROUP (CIAO) ALLOW YOU TO CHOOSE MINIMUM COUNTS/BIN, OR SPECIFIED NO. OF CHANNELS/BIN OVER GIVEN RANGES.

- I PREFER TO MAKE THESE CHOICES DURING ANALYSIS, WITH A MORE FLEXIBLE SET OF CRITERIA.
GROUPING & NOTICING

SUZAKU CYG X-1 - NO GROUPING

S/N > 8 & HWHM RES.
GROUPING & NOTICING

- **THERE ARE TWO MAIN FUNCTIONS TO USE IN ISIS:**
  
  isis> group(pca;min_sn=5,minChan=1,bounds=3,unit="kev");
  isis> notice_values(pca,3,20;unit="kev");

- **GROUP CAN GET FANCY FOR DATA ON THE SAME GRID:**
  
  isis> group([1,2];min_sn={5,8},minChan={1,4},
              bounds={3,15},unit="kev",sn_data=1);
  isis> notice_values([1,2],3,5,8,20;unit="kev");

- **EXCLUDE/INCLUDE DATA WITHOUT CHANGING BINS**
  
  isis> exclude(1);  % ignore dataset #1
  isis> include(1);  % bring it back with its previous bins