Inspired by Maurice, I'm re-reducing the zeta Pup observation; using his script as a guide – but not actually using the script; At each step, I'll comment on whether I'm doing something different than he did

Note: using CIAO v3.3

```
[cohen@ruby reproc]$ pwd
/home/cohen/chandra/zpup/reproc

[cohen@ruby reproc]$ ls
acisf00640_000N003_bpix1.fits  acisf00640_000N003_flt1.fits
pcadf070693006N003_asol1.fits
acisf00640_000N003_evt1.fits   pcadf070637951N003_asol1.fits
```

http://cxc.harvard.edu/ciao/threads/acisdetectafterglow/

```
[cohen@ruby reproc]$ punlearn dmtcalc
[cohen@ruby reproc]$ pset dmtcalc infile=acisf00640_000N003_evt1.fits
[cohen@ruby reproc]$ pset dmtcalc outfile=acis640_reset_evt1.fits
[cohen@ruby reproc]$ pset dmtcalc
expression="status=status,status=X15F,status=X14F,status=X13F,status=X12F"
[cohen@ruby reproc]$ dmtcalc
```

Note: the working evt1 file is now:

acis640_reset_evt1.fits

http://cxc.harvard.edu/ciao/threads/acishotpixels/

```
[cohen@ruby reproc]$ dmkeypar acis640_reset_evt1.fits READMODE echo+
TIMED
[cohen@ruby reproc]$ dmkeypar acis640_reset_evt1.fits DATAMODE echo+
FAINT
```

The thread says "This is a TIMED FAINT mode observation, so `acis_run_hotpix` may be used."

The thread also says:

"Since this data has already had `acis_detect_afterglow` correction removed, the `ASCDSVER` keyword value is the CIAO software version that was used. If you have not removed the correction yet, you should see something like this:

```
unix% dmkeypar acisf00459_000N002_evt1.fits ASCDSVER echo+
R4CU5UPD11.1"
```

So, this confirms what we did in the very first step – removing the afterglow correction

Next we'll create a new bad pixels file:

Using: **acis_run_hotpix**

Files needed: evt1; bpix1; msk1; bias0; pbk0
This seems to include the *old/original* bpix file; the msk, pbk, and bias files (six of the latter) are all in the secondary/ directory and have now been copied into reproc/

Have to make a list of the bias files, which will then be input to this task:

```
[cohen@ruby reproc]$ emacs bias_files.lis &
```

```
acisf070637510N003_0_bias0.fits
acisf070637510N003_1_bias0.fits
acisf070637510N003_2_bias0.fits
acisf070637510N003_3_bias0.fits
acisf070637510N003_4_bias0.fits
acisf070637510N003_5_bias0.fits
```

Setting the parameters:

```
[cohen@ruby reproc]$ punlearn acis_run_hotpix
[cohen@ruby reproc]$ pset acis_run_hotpix infile=acis640_reset_evt1.fits
[cohen@ruby reproc]$ pset acis_run_hotpix outfile=acis640_new_bpix1.fits
[cohen@ruby reproc]$ pset acis_run_hotpix badpixfile=acisf00640_000N003_bpix1.fits
[cohen@ruby reproc]$ pset acis_run_hotpix biasfile=@bias_files.lis
[cohen@ruby reproc]$ pset acis_run_hotpix maskfile=acisf00640_000N003_msk1.fits
[cohen@ruby reproc]$ pset acis_run_hotpix pbkfile=acisf070638815N003_pbk0.fits
```

```
[cohen@ruby reproc]$ acis_run_hotpix
Input event list (acis640_reset_evt1.fits):
Output bad pixel file (acis640_new_bpix1.fits):
Input parameter block file (acisf070638815N003_pbk0.fits):
Input mask file ( <filename> | none | NONE ) (acisf00640_000N003_msk1.fits):
Input bad pixel file (acisf00640_000N003_bpix1.fits):
Input bias images (@bias_files.lis):
[cohen@ruby reproc]$
```

Now we have to apply this newly created bad pixel file using **acis_process_events**:

```
[cohen@ruby reproc]$ pset acis_process_events badpixfile=acis640_new_bpix1.fits
```

The thread also directs us to set this observation-specific bad pixel file for "any subsequent analysis task" -

http://cxc.harvard.edu/ciao/threads/badpix/

```
[cohen@ruby reproc]$ punlearn ardlib
[cohen@ruby reproc]$ acis_set_ardlib acis640_new_bpix1.fits
Warning:  found possibly out of date user parameter file, renamed to
/home/cohen/pfiles/acis_set_ardlib_20061216.14:55:35.par
Updated ardlib parameter file: /home/cohen/pfiles/ardlib.par
  AXAF_ACIS0_BADPIX_FILE -> CALDB
  AXAF_ACIS1_BADPIX_FILE -> CALDB
  AXAF_ACIS2_BADPIX_FILE -> CALDB
  AXAF_ACIS3_BADPIX_FILE -> CALDB
  AXAF_ACIS4_BADPIX_FILE ->
/home/cohen/chandra/zpup/reproc/acis640_new_bpix1.fits[BADPIX4]
  AXAF_ACIS5_BADPIX_FILE ->
/home/cohen/chandra/zpup/reproc/acis640_new_bpix1.fits[BADPIX5]
  AXAF_ACIS6_BADPIX_FILE ->
/home/cohen/chandra/zpup/reproc/acis640_new_bpix1.fits[BADPIX6]
  AXAF_ACIS7_BADPIX_FILE ->
/home/cohen/chandra/zpup/reproc/acis640_new_bpix1.fits[BADPIX7]
  AXAF_ACIS8_BADPIX_FILE ->
/home/cohen/chandra/zpup/reproc/acis640_new_bpix1.fits[BADPIX8]
  AXAF_ACIS9_BADPIX_FILE ->
/home/cohen/chandra/zpup/reproc/acis640_new_bpix1.fits[BADPIX9]
```

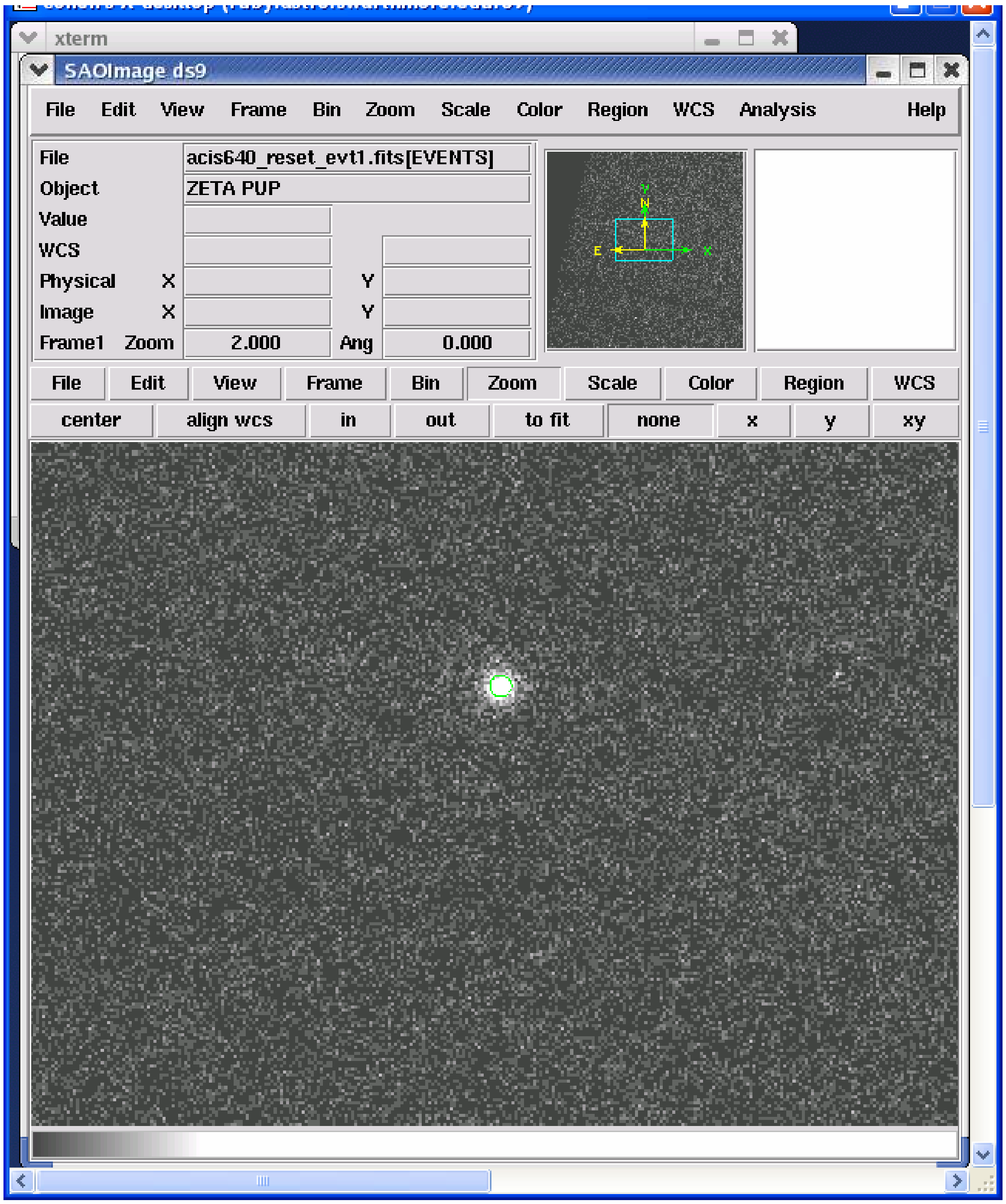
Now to the main script for generating dispersed spectra and associated files:

http://cxc.harvard.edu/ciao/threads/spectra_hetgacis/

First task is **tgdetect**

```
[cohen@ruby reproc]$ punlearn tgdetect
[cohen@ruby reproc]$ pset tgdetect infile=acis640_reset_evt1.fits
[cohen@ruby reproc]$ pset tgdetect outfile=acis640_reset_src1a.fits
[cohen@ruby reproc]$ tgdetect
Input L1 event file (acis640_reset_evt1.fits):
Input source position(s) file from previous OBI or NONE (NONE):
Output source position(s) file name (acis640_reset_src1a.fits):
# DMCOPY (CIAO3.3): Filter data type mismatch for DOUBLE
```

the thread says to ignore this warning


xterm
SAOImage ds9
File Edit View Frame Bin Zoom Scale Color Region WCS Analysis Help
File
Object
Value
WCS
Physical X Y
Image X Y
Frame1 Zoom
Ang
File Edit View Frame Bin Zoom Scale Color Region WCS
center align wcs in out to fit none x y xy

File acis640_reset_evt1.fits[EVENTS]
Object ZETA PUP
Frame1 Zoom 2.000 Ang 0.000

The source detection seemed to go fine.

The next step is to get the region mask, using **tg_create_mask**:

```
[cohen@ruby reproc]$ punlearn tg_create_mask
[cohen@ruby reproc]$ pset tg_create_mask infile=acis640_reset_evt1.fits
[cohen@ruby reproc]$ pset tg_create_mask outfile=acis640_reset_evt1_L1a.fits
[cohen@ruby reproc]$ pset tg_create_mask input_pos_tab=acis640_reset_src1a.fits
```

```
[cohen@ruby reproc]$ tg_create_mask
Input event file or stack (acis640_reset_evt1.fits):
Output region file or stack (acis640_reset_evt1_L1a.fits):
Input table with zero order positions or stack (acis640_reset_src1a.fits):
Observed grating type (header_value|HETG|HEG|MEG|LETG)
(HETG|HEG|MEG|LETG|header_value|HEADER_VALUE) (header_value):
```

The next step is to assign grating events to spectral orders using **tg_resolve_events**:

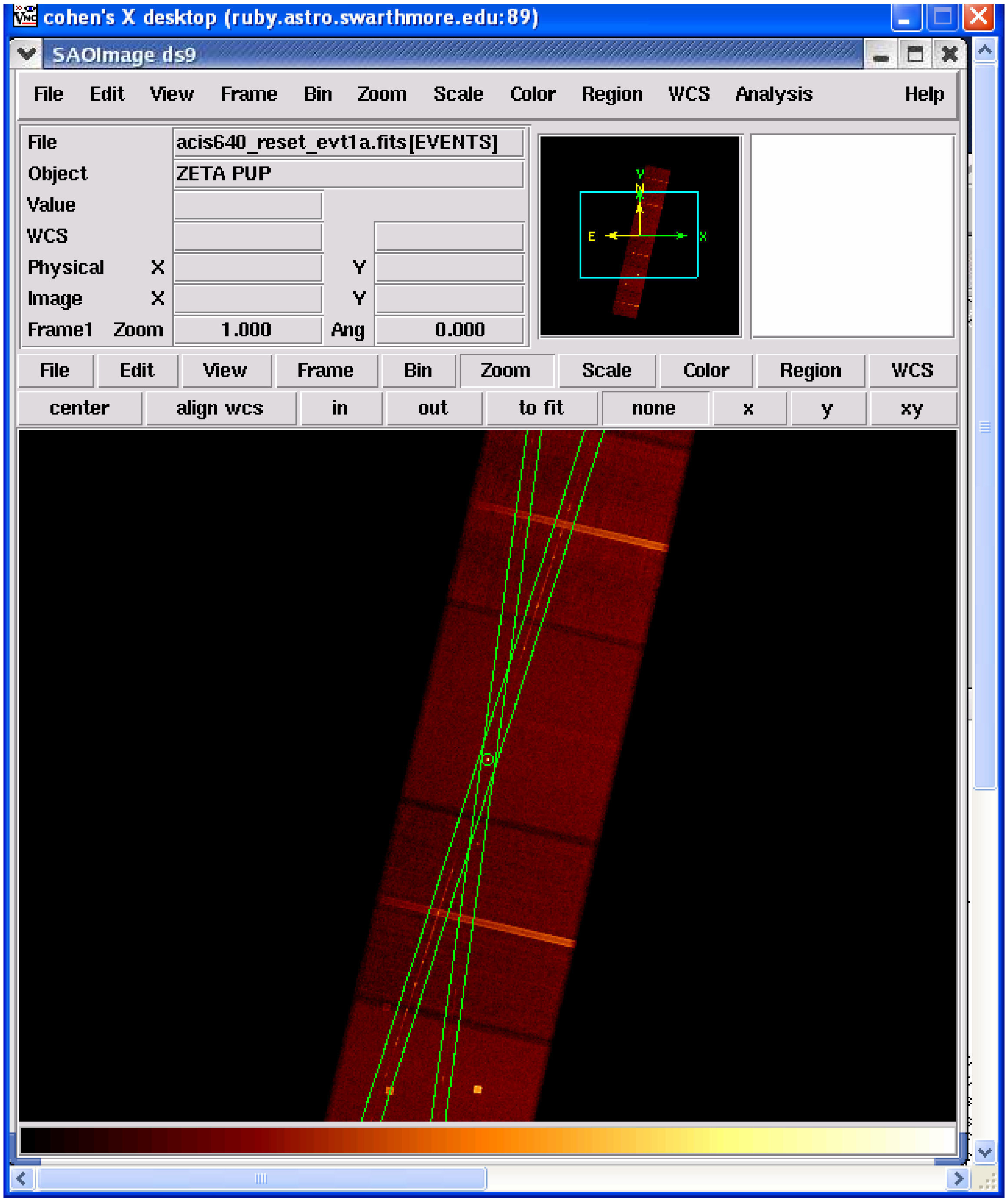
First, we have to copy in the *asol1* files, and make a file listing them (which will be used as an input parameter to this task).

```
[cohen@ruby reproc]$ ls *asol*
pcadf070637951N003_asol1.fits  pcadf070693006N003_asol1.fits
[cohen@ruby reproc]$ emacs pcad_asol1.lis &
[cohen@ruby reproc]$ more pcad_asol1.lis
pcadf070637951N003_asol1.fits
pcadf070693006N003_asol1.fits

[cohen@ruby reproc]$ punlearn tg_resolve_events
[cohen@ruby reproc]$ pset tg_resolve_events infile=acis640_reset_evt1.fits
[cohen@ruby reproc]$ pset tg_resolve_events outfile=acis640_reset_evt1a.fits
[cohen@ruby reproc]$ pset tg_resolve_events regionfile=acis640_reset_evt1_L1a.fits
[cohen@ruby reproc]$ pset tg_resolve_events acaofffile="@pcad_asol1.lis"
[cohen@ruby reproc]$ tg_resolve_events
Input event file or stack (acis640_reset_evt1.fits):
Input region file or stack (acis640_reset_evt1_L1a.fits):
Output event file or stack (acis640_reset_evt1a.fits):
Input aspect offset file (@pcad_asol1.lis):
# tg_resolve_events (CIAO3.3): The following error occurred 3401830 times:
        dsTREUNKNOWNINCOLERR -- WARNING: Not loading data from
unrecognized level 1.5 input column.
```

Note that the thread says that this warning can be safely ignored.

Looking at the results:


cohen's X desktop (ruby.astro.swarthmore.edu:89)
SAOImage ds9
File Edit View Frame Bin Zoom Scale Color Region WCS Analysis Help
File
Object
Value
WCS
Physical X Y
Image X Y
Frame1 Zoom Ang
File Edit View Frame Bin Zoom Scale Color Region WCS
center align wcs in out to fit none x y xy

File: acis640_reset_evt1a.fits[EVENTS]
Object: ZETA PUP
Frame1 Zoom: 1.000 Ang: 0.000

Looks good.

Now we move on to generating a new level 2 events table:

```
[cohen@ruby reproc]$ punlearn dmcopy
```

```
[cohen@ruby reproc]$ dmcopy
"acis640_reset_evt1a.fits[EVENTS][grade=0,2,3,4,6,status=0]"
acis640_reset_flt1_evt1a.fits opt=all
```

Next we apply the GTI filters:

```
[cohen@ruby reproc]$ punlearn dmcopy
[cohen@ruby reproc]$ dmcopy
"acis640_reset_flt1_evt1a.fits[EVENTS][@acisf00640_000N003_flt1.fits][cols -phas]"
acis640_reset_flt1_evt2.fits opt=all
```

And next, we destreak chip S4:

```
[cohen@ruby reproc]$ punlearn destreak
[cohen@ruby reproc]$ pset destreak infile=acis640_reset_flt1_evt2.fits
[cohen@ruby reproc]$ pset destreak outfile=acis640_reset_flt1_dstrk_evt2.fits
[cohen@ruby reproc]$ destreak
Input dataset/block specification (acis640_reset_flt1_evt2.fits):
Output dataset/block specification (acis640_reset_flt1_dstrk_evt2.fits):
```

And now we can extract the grating spectra:

```
[cohen@ruby reproc]$ punlearn tgextract
[cohen@ruby reproc]$ pset tgextract infile=acis640_reset_flt1_dstrk_evt2.fits
[cohen@ruby reproc]$ pset tgextract outfile=acis640_pha2.fits
[cohen@ruby reproc]$ tgextract
Input event file (output event file from L1.5 processing)
(acis640_reset_flt1_dstrk_evt2.fits):
If typeII, enter full output file name or '.'; if typeI, enter output rootname
(acis640_pha2.fits):
Input ancillary response file name (none):
Input redistribution file name (none):
Source ID's to process: 'all', comma list, @file (all):
Grating parts to process: HETG, HEG, MEG, LETG, header_value
(HETG|HEG|MEG|LETG|header_value) (header_value):
Grating diffraction orders to process: 'default', comma list, range list, @file (default):
Output file type: typeI (single spectrum) or typeII (multiple spectra)
(pha_typeI|pha_typeII) (pha_typeII):
```

→ OK, the new pha2 file has been created.

To work with it, we have to make rmfs and grating arfs:

Copied over 8 calibration files from the $CALDB directory:
/amundsen/opt/caldb_3.2.2/data/chandra/tel/grating/hetg/cpf/lsf

http://cxc.harvard.edu/ciao/threads/mkgrmf_aciss/

We'll make files (both gARFs and rmfs) for only the first order spectra

For the HEG+1 order:

```
[cohen@ruby reproc]$ punlearn mkgrmf
[cohen@ruby reproc]$ pset mkgrmf order=1
[cohen@ruby reproc]$ pset mkgrmf grating_arm=HEG
[cohen@ruby reproc]$ pset mkgrmf outfile=heg_p1.rmf
[cohen@ruby reproc]$ pset mkgrmf obsfile="acis640_pha2.fits[SPECTRUM]"
[cohen@ruby reproc]$ pset mkgrmf regionfile=acis640_pha2.fits
[cohen@ruby reproc]$ pset mkgrmf detsubsys=ACIS-S3
[cohen@ruby reproc]$ pset mkgrmf wvgrid_arf=compute
[cohen@ruby reproc]$ pset mkgrmf wvgrid_chan=compute
[cohen@ruby reproc]$ pset mkgrmf clobber=no
[cohen@ruby reproc]$ mkgrmf
Output File Name (heg_p1.rmf):
Enter ARF side wavelegth grid [angstroms] (compute):
Enter channel-side wavelegth grid [angstroms] (compute):
Enter Grating order (1):
Name of fits file with obs info (acis640_pha2.fits[SPECTRUM]):
File containing extraction region (acis640_pha2.fits):
SrcID (1):
Enter RMF threshold (1e-06):
Verbosity (0:5) (0):
Detector Name (e.g., ACIS-S3) (ACIS-S3):
Enter Grating Arm (HEG|MEG|LEG|NONE) (HEG):
```

OK, now making the HEG-1 order:
(just need to change a couple of parameters)

```
[cohen@ruby reproc]$ pset mkgrmf order=-1
[cohen@ruby reproc]$ pset mkgrmf outfile=heg_m1.rmf
```

Now on to the MEG-1 order:

```
[cohen@ruby reproc]$ pset mkgrmf grating_arm=MEG
[cohen@ruby reproc]$ pset mkgrmf outfile=meg_m1.rmf
```

And finally, the MEG+1 order:

```
[cohen@ruby reproc]$ pset mkgrmf order=1
```

```
[cohen@ruby reproc]$ pset mkgrmf outfile=meg_p1.rmf
```

Now, making the grating ARFs:

http://cxc.harvard.edu/ciao/threads/mkgarf_hetgacis/

```
[cohen@ruby reproc]$ pset mkgarf maskfile=acisf00640_000N003_msk1.fits

[cohen@ruby reproc]$ punlearn fullgarf
Warning:  found possibly out of date user parameter file, renamed to
/home/cohen/cxcds_param/fullgarf_20061218.11:55:53.par
[cohen@ruby reproc]$ pset fullgarf phafile=acis640_pha2.fits
[cohen@ruby reproc]$ pset fullgarf pharow=3
[cohen@ruby reproc]$ pset fullgarf evtfile=acis640_reset_flt1_dstrk_evt2.fits
[cohen@ruby reproc]$ pset fullgarf asol=@pcad_asol1.lis
[cohen@ruby reproc]$ pset fullgarf engrid="grid(heg_m1.rmf[cols
ENERG_LO,ENERG_HI])"
[cohen@ruby reproc]$ pset fullgarf dtffile=")evtfile"
[cohen@ruby reproc]$ pset fullgarf badpix=acis640_new_bpix1.fits
[cohen@ruby reproc]$ pset fullgarf rootname=acis640

…

Finished processing grating arfs for ccd_id= 4 5 6 7

dmarfadd @thisfile.istemporary acis640HEG_-1_garf.fits

/amundsen/opt/ciao_3.3/contrib/bin/fullgarf finished.
```

Now, making the other three:

```
[cohen@ruby reproc]$ pset fullgarf pharow=4
[cohen@ruby reproc]$ pset fullgarf engrid="grid(heg_p1.rmf[cols
ENERG_LO,ENERG_HI])"
[cohen@ruby reproc]$ fullgarf
```

OK, made the last two as well. A few error messages, but they look harmless, and lots of intermediate files strewn in my directory.

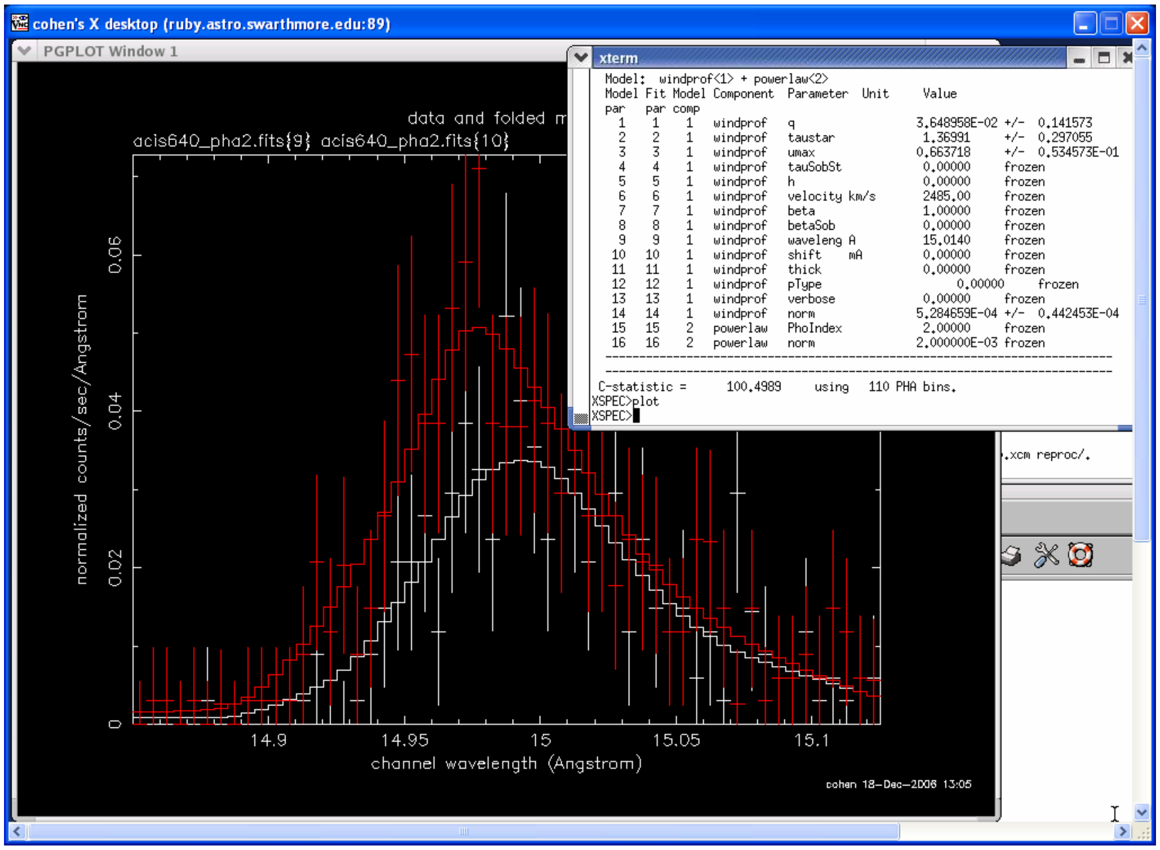
But now I have rmfs and garfs for all four first order grating arms.

Let’s examine the data as we had before:

Oh, one last thing: have to make a background pha file for XSPEC analysis.

```
[cohen@ruby reproc]$ tg_bkg acis640_pha2.fits
Input file has been changed, BACKFILE keyword has been added
```

It looks like I'm now getting results more like Maurice's. Here's my best-fit windprof model to the MEG+/-1 Fe XVII 15.014 line:


cohen's X desktop (ruby.astro.swarthmore.edu:89)
PGPLOT Window 1
data and folded m
acis640_pha2.fits{9} acis640_pha2.fits{10}
normalized counts/sec/Angstrom
0
0.02
0.04
0.06
14.9
14.95
15
15.05
15.1
channel wavelength (Angstrom)
cohen 18-Dec-2006 13:05
xterm
.xcm reproc/.

```
 Model:  windprof<1> + powerlaw<2>
 Model Fit Model Component  Parameter  Unit     Value
 par   par comp
   1    1    1   windprof   q                  3.648958E-02 +/-  0.141573
   2    2    1   windprof   taustar             1.36991     +/-  0.297055
   3    3    1   windprof   umax               0.663718     +/-  0.534573E-01
   4    4    1   windprof   tauSobSt            0.00000     frozen
   5    5    1   windprof   h                   0.00000     frozen
   6    6    1   windprof   velocity km/s       2485.00     frozen
   7    7    1   windprof   beta                1.00000     frozen
   8    8    1   windprof   betaSob             0.00000     frozen
   9    9    1   windprof   waveleng A          15.0140     frozen
  10   10    1   windprof   shift    mA         0.00000     frozen
  11   11    1   windprof   thick               0.00000     frozen
  12   12    1   windprof   pType                    0.00000     frozen
  13   13    1   windprof   verbose             0.00000     frozen
  14   14    1   windprof   norm               5.284659E-04 +/-  0.442453E-04
  15   15    2   powerlaw   PhoIndex            2.00000     frozen
  16   16    2   powerlaw   norm               2.000000E-03 frozen
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 C-statistic =      100.4989     using   110 PHA bins.
XSPEC>plot
XSPEC>
```