Tutorial on Data Analysis

Covering:

Swift (BAT, XRT, UVOT)

XMM-Newton SAS

XSPEC

Kim Page

(UKSSDC, University of Leicester)

Introduction

Currently there are many satellite missions which can observe GRBs, each of which produces data which need to be analysed differently.

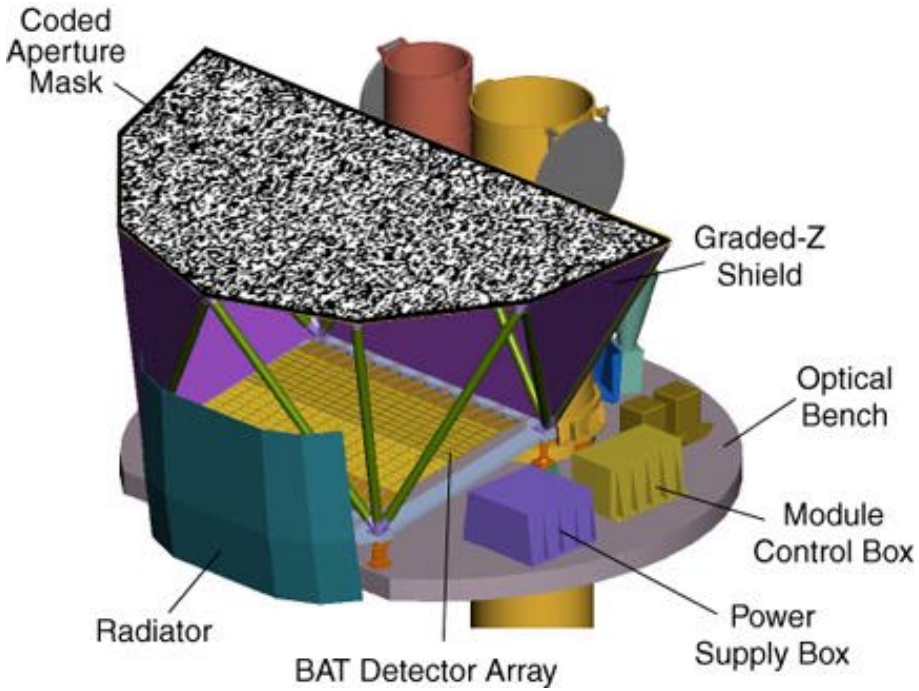
In this tutorial we will cover Swift and XMM-Newton data reduction and extraction, followed by a brief introduction to the spectral fitting programme XSPEC.

This will be something of a whirlwind tour unfortunately – 3 hours is not long enough!!

Swift - BAT

Burst Alert Telescope

The BAT is a coded-aperture mask instrument: that is, it images hard X-rays (15-350 keV), but it does not focus them. It has a high sensitivity and a large field of view (2 steradian).



The mask consists of ~54000 lead tiles in a random pattern, while there are 32 768 individual CdZnTe elements in the detector plane.

The BAT constantly monitors the sky, searching for rate and/or image increases. When a rate increase is detected, a follow-up image increase (i.e. a point source) is also looked for. Initial image (rather than rate) triggers may be indicative of high-z bursts, because of time dilation.

Mask-weighting

Mask-weighting is a way of background-subtracting coded-mask data. Although the files in the archives are likely to be mask-weighted, the software may have improved since then. Also, since mask-weighting depends on the position of the source, refined RA/Dec values means mask-weighting should be redone.

```
>batmaskwtevt detmask=[obsid]/bat/hk/sw[obsid]bdqcb.hk
Input event file name: [obsid]/bat/event/sw[obsid]bevshsp_uf.evt
Input attitude file name: [obsid]/auxil/sw[obsid]sat.fits
Input RA: www.xxx      Must be in decimal degrees!
Input Dec: yyy.zzz     Must be in decimal degrees!
```

This does not produce a new file, but rather adds (or alters) a column called MASK_WEIGHT in the event file. This means the *uf.evt file must not be gzipped.

If the detector mask (map of excluded detectors, since they aren't always all on) is missing, one can be made using bathotpix. The *bdqcb.hk file is labelled as "Map of excluded detectors" in the archive.

Images - 1

First, make a detector plane image (DPI).

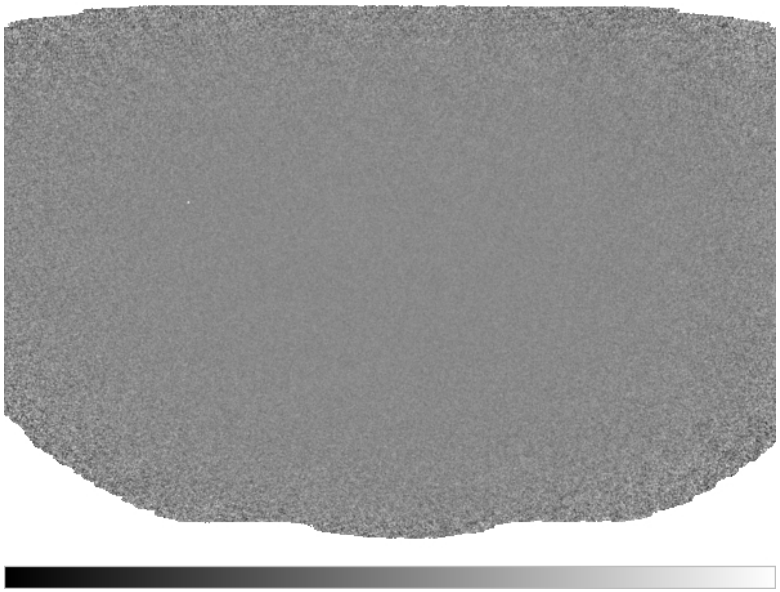
>batbinevt detmask=[obsid]/bat/hk/sw[obsid]bdqcb.hk
Input event file name: [obsid]/bat/event/sw[obsid]bevshsp_uf.evt
Output file name: grb.dpi
Make light-curve or spectrum: dpi
Histogram time bin size: 0
Time binning algorithm: u
Energy bin list: 15-350

Alternatively:
Energy bin list: 15-25, 25-50, 50-100, 100-350
if you would like to create a file with 4 different energy extensions.

Images - 2

Then, perform a Fourier Transform to obtain a sky image:

```
>batfftimage
Input dpi file name: grb.dpi
Output image name: grb.img
Input attitude file name: [obsid]/auxil/sw[obsid]sat.fits
```



Source Detection

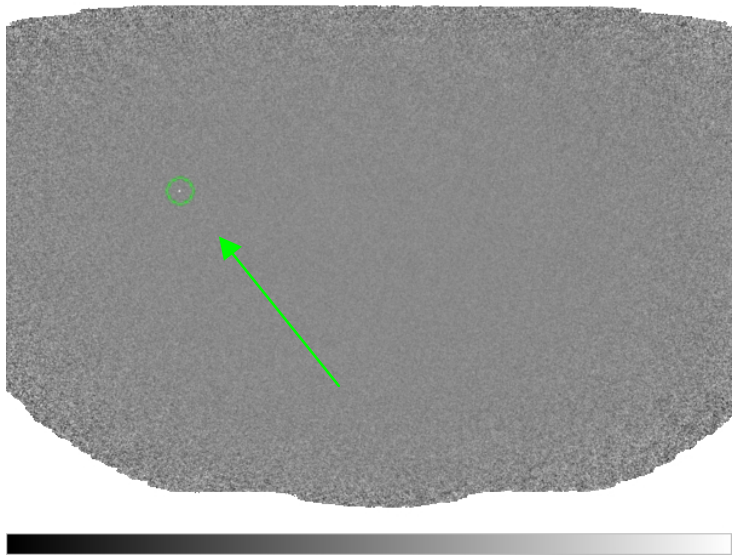
To detect sources, run the simple task batcelldetect.

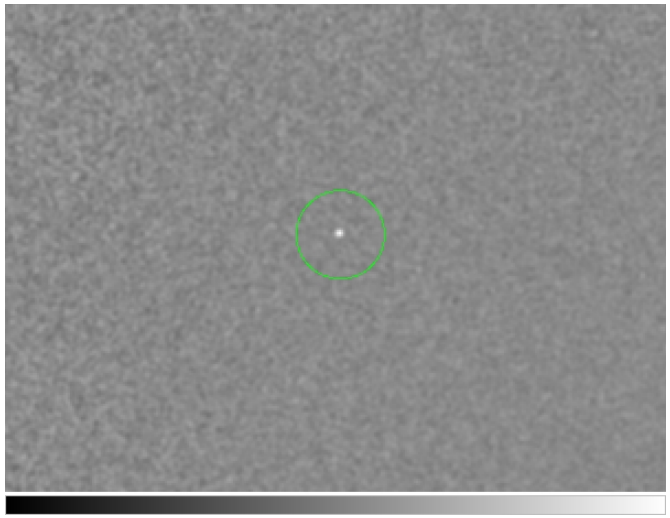
>batcelldetect

Input sky image file name: grb.img

Output source list: sources.fits

Pixel signal to noise detection threshold: 6





Light-curves - 1

The task batbinevt is multi-purpose. We used it to extract a DPI earlier, and can repeat the method to make light-curves and spectra:

```
>batbinevt detmask=[obsid]/bat/hk/sw[obsid]bdqcb.hk
Input event file name: [obsid]/bat/event/sw[obsid]bevshsp_uf.evt
Output file name: grb.lc
Make light-curve or spectrum: lc
Histogram time bin size: 1.0
Time binning algorithm: u
Energy bin list: 15-350
```

Again, a comma-separated list can be given for the energy bins.

If you want to plot the light-curves against time since trigger (rather than MET), use fcalc, giving TIME-TRIGTIME as the arithmetic expression required.

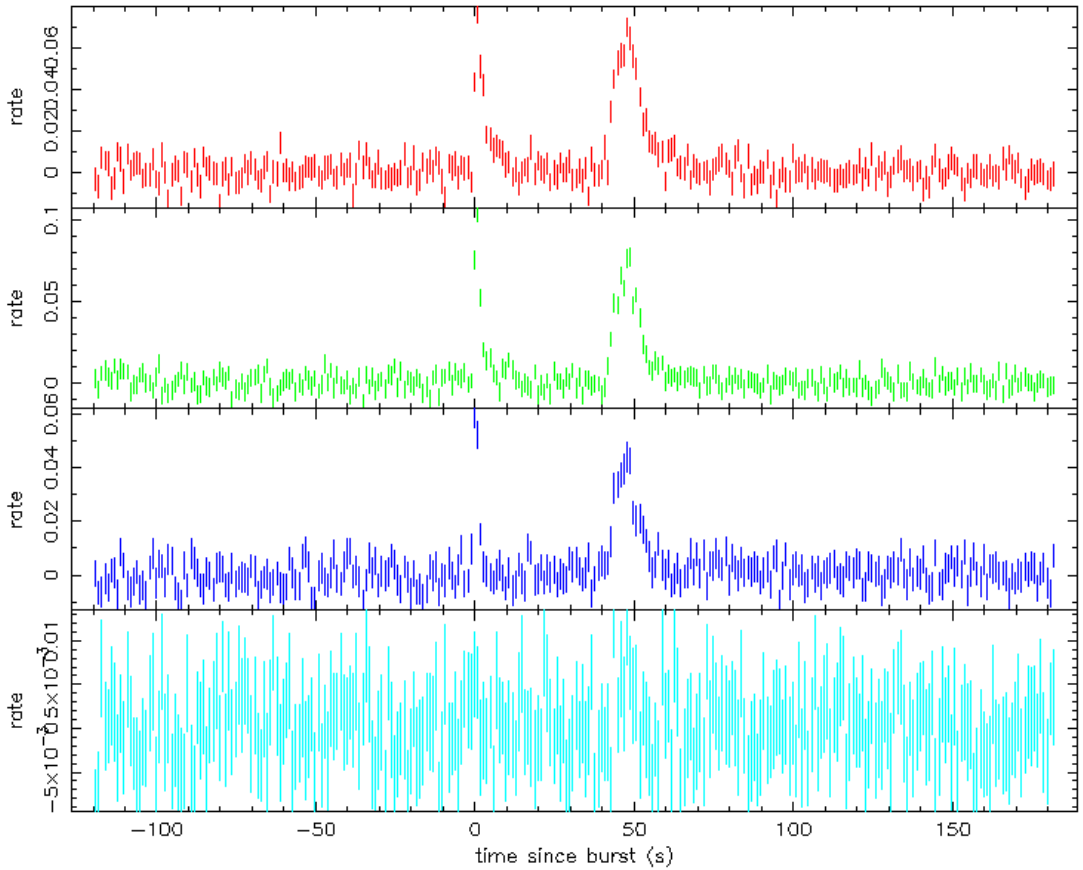
Light-curves - 2

If you have a multi-vector light-curve (i.e., more than 1 energy band), they can be plotted as follows:

```
>fplot 4channel.lc
Name of X Axis Parameter[error]: TIME
Name of Y Axis Parameter[error]: RATE(1)[ERROR], RATE(2)[ERROR],
RATE(3)[ERROR], RATE(4)[ERROR]
Lists of rows: -
Device: /xw
```

and you get…

Light-curves - 3



15-25 keV

25-50 keV

50-100 keV

100-350 keV

Timing

To determine timing properties of the burst, battblocks is used:

> battblocks durfile=dur.gti txx=67.0 bkgsub=yes countscol=rate
Input data file name: grb.lc
Output GTI file name: bb.gti

txx=67.0 allows the user to choose another period to determine – in this case, the time during which 67% of the total counts are detected.

dur.gti will contain the start and stop times (in MET) for T_{90}, T_{50}, T_{XX}, 1 second peak flux, the total period covered and the intervals used for background subtraction. Without including this on the command line, the numbers are just written to the screen.

bb.gti lists the Baysian block periods – i.e. time intervals of "constant" flux.

Use countscol=TOT_RATE if using a TDRSS light-curve.

Spectra - 1

>batbinevt detmask=[obsid]/hk/sw[obsid]bdqcb.hk
Input event file name: [obsid]/bat/event/sw[obsid]bevshsp_uf.evt
Output file name: grb.pha
Make light-curve or spectrum: pha
Histogram time bin size: 0
Time binning algorithm: u
Energy bin list: CALDB:80

CALDB:80 uses the "standard" 80 channels which are defined in the Calibration Database.
If you want a spectrum for a specific time interval, include the tstart/tstop parameters on the command line; these must be given in terms of MET, e.g.

>batbinevt detmask=[obsid]/hk/sw[obsid]bdqcb.hk tstart=295003248 tstop=295003312

Spectra - 2

A systematic error vector must be applied to the BAT spectra before use to account for residuals in the response matrix; this is done using batphasyserr. batupdatephakw then needs to be run to ensure that the position of the burst in instrument coordinates is known (since the spacecraft slews, this can change, but the knowledge is required for the flux determination).

```
>batphasyserr
Input spectrum filename: grb.pha
Systematic error filename: CALDB   NB CALDB must be in capitals!
>batupdatephakw
Input spectrum filename: grb.pha
Auxiliary raytracing filename: [obsid]/bat/event/sw[obsid]bevtr.fits
```

N.B. Early datasets may not contain this *bevtr.fits file. If the file is missing, batmaskwtevt needs to be run to produce one: include auxfile=sw[obsid]bevtr.fits on the batmaskwtevt command line to output the ray-tracing results to a file.

Spectra - 3

Finally, a detector response matrix must be built, so that the spectrum can be plotted in terms of energy:

```
>batdrmgen
Input event PHA file name: grb.pha
Output response matrix file name: grb.rsp
DAP housekeeping file name: none
```

The BAT team recommend fitting the spectra between 15-150 keV, since the BAT mask becomes transparent around 150 keV. Data below 14 keV and above 195 keV should definitely be ignored.

See Sakamoto et al. (2008, ApJS, 175, 179) for information about weighting RMFs.

Things to check - 1

Older data may not have been processed using the most up-to-date energy calibration. In this case, it may be a good idea to (re-)run bateconvert. To check whether this is necessary:

```
>fkeyprint sw[obsid]bevshsp_uf.evt GAIN
```

This will show various results, including:

```
GAINAPP =                    T / Gain correction has been applied
GAINMETH= 'FIXEDDAC'         / Cubic ground gain/offset correction using
                                DAC-b
```

If the keywords don't exist, or are NOT set to True and FIXEDDAC respectively, bateconvert should be run.

```
>bateconvert
Input file name: [obsid]/bat/event/sw[obsid]bevshsp_uf.evt
Flight calibration (gain/offset) file name: [obsid]/bat/hk/sw[obsid]bgocb.hk
Quadratic pulser residuals calibration file name: CALDB
Ground pulser DAC to keV calibration file name: CALDB
Flight pulser DAC to keV calibration file name: CALDB
```

Things to check - 2

If no detector mask (*bdqcb.hk) can be found, produce a DPI using batbinevt (but without the detmask= command!) followed by:

> bathotpix detmask=[obsid]/bat/hk/sw[obsid]bdecb.fits
Input detector plane image: grb.dpi
Output image mask: grb.mask

Then grb.mask can be used in place of the *bdqcb.hk file referred to in earlier examples.

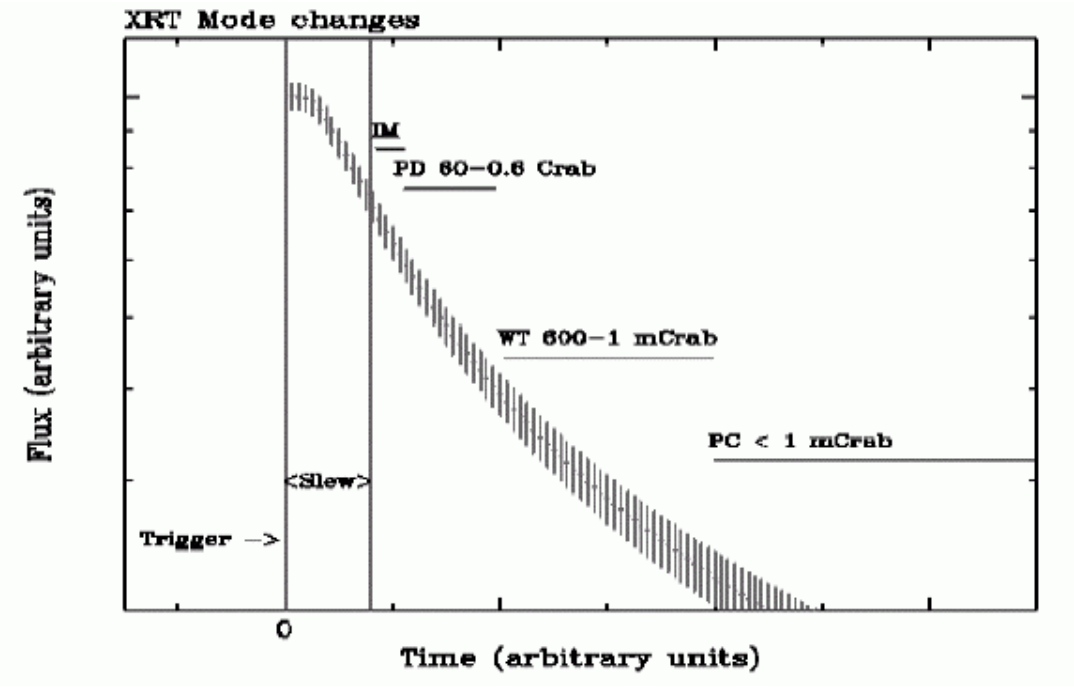
Finally…

The "normal" processing run for BAT GRBs (standard light-curves, spectra etc.) can be performed simply by running batgrbproduct and giving an input directory containing the auxil, bat and tdrss subdirectories. This incorporates the steps covered in the previous slides.

Much easier! ☺

Swift - XRT

X-Ray Telescope



- Imaging mode
  - Exposure of 0.1 or 2.5 s
  - No spectroscopy data
- Photo-Diode mode (Low-Rate and Piled-Up) – Currently disabled
  - No spatial information
  - High time resolution (0.14 ms)
- Windowed Timing mode
  - 1-dimensional imaging
  - 1.7 ms time resolution
- Photon Counting mode
  - 2.5 s time resolution

Running the pipeline

The pipeline requires the auxil and xrt directories to run.

xrtpipeline cleanup=no >& log

- cleanup=no keeps all the temporary files the pipeline produces; this is mainly useful because it keeps the extraction region used, which is centred on the position given.
- clobber=no is the default – files will not be overwritten.
- createexpomap=yes could be useful – see later slide about exposure maps. This will become the default in the next release of the software.
- The gti expression includes (ELV>=30||BR_EARTH>=120)&&(SUN_ANGLE>=45 &&ANG_DIST<=0.08&&MOON_ANGLE>=14)&&(CCDTemp>=-102&&CCDTemp<=-47). To change this, include gtiexpr="expression" on the command line. (Usually the default expression is fine, though.)

xrtpipeline example

```
==================================================================
            Running SWIFT XRT pipeline
 Task: xrtpipeline Version: 0.12.4 Release Date: 2009-08-25
==================================================================
Source RA position (POINT to use optical axis direction) (degrees or hh mm ss.s) [ ] :
13 25 27.6

Source DEC position (POINT to use optical axis direction) (degrees or dd mm ss.s)[ ] :
-43 01 09

Target Archive Directory Path [ ] : 00050950004

Stem for FITS input files [i.e. sw00000000000] [ ] :sw00050950004

Directory for outputs [ ] : 00050950004-xrt
```

Hints & Tips

- Sometimes more data can be obtained for an observation by relaxing the observing constraints – specifically the (ELV>=30||BR_EARTH>=120) part. This is more relevant for GRBs than ToOs.
- If the count-rate within the field of view is close to the PC/WT boundary, mode-switching can occur. This means that, even if you expect your object to be in PC mode, a substantial portion can end up in WT event files. Time is also lost each time the mode changes. This is annoying, but there's nothing that you, as an observer, can do about it. ☹
- Beware of pile-up! If the count-rate for the source is more than ~0.6-0.7 count s^{-1}, the PC data are likely to suffer from pile-up. A piled-up spectrum will be flatter/harder than one which has the pile-up correctly accounted for. A method for estimating the amount of pile-up will be covered later.

XSELECT - 1

XSELECT is the easiest way to extract X-ray images, spectra, light-curves:

> xselect

Enter session name >[xsel]

> read event

Enter the Event file dir >[ ] ./

Enter Event file list > [ ] sw[obsid]xpcw3po_cl.evt

Got new mission: SWIFT

Reset the mission? >[ ] yes

Note that some PC eventlists will be w2 or w4 rather than w3. This signifies the size of the window and the default has changed over time. WT data will be sw[obsid]xwtw2po_cl.evt.

By default, PC data are grades 0-12 and WT are 0-2. The modes are also calibrated for grade 0 alone, though (filter grade 0 in XSELECT).

XSELECT - 2

> extract image

```
Total      Good       Bad: Region     Time     Phase     Grade     Cut
 6258     6258                  0          0          0          0       0
==================================================================
Grand Total      Good   Bad:  Region  Time  Phase  Grade  Cut
 6258             6258                 0        0        0        0     0
   in  3603.8     seconds
 Image          has  6258 counts for  1.737     counts/sec
```

> plot image – opens in ds9. Source and background regions can be defined and saved.

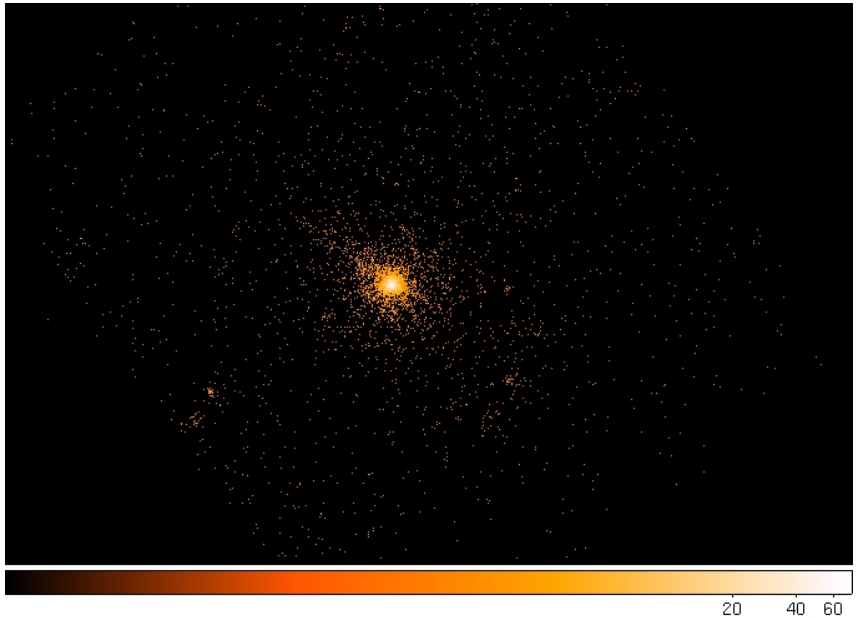
> filter region src.reg – clear region removes this filtering

> extract curve – light-curve produced for whatever filtering has been defined

> plot curve

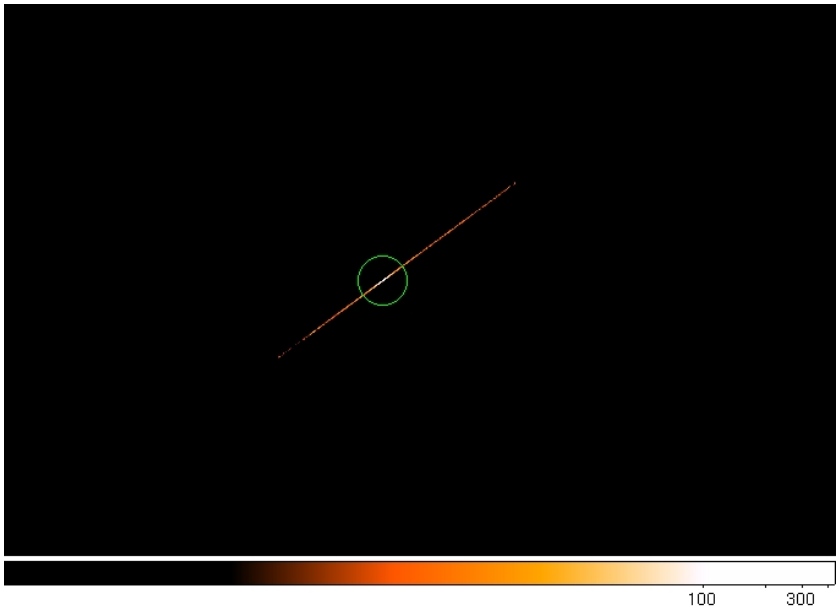
> save curve – N.B. XSELECT does not save anything by default.

XSELECT images

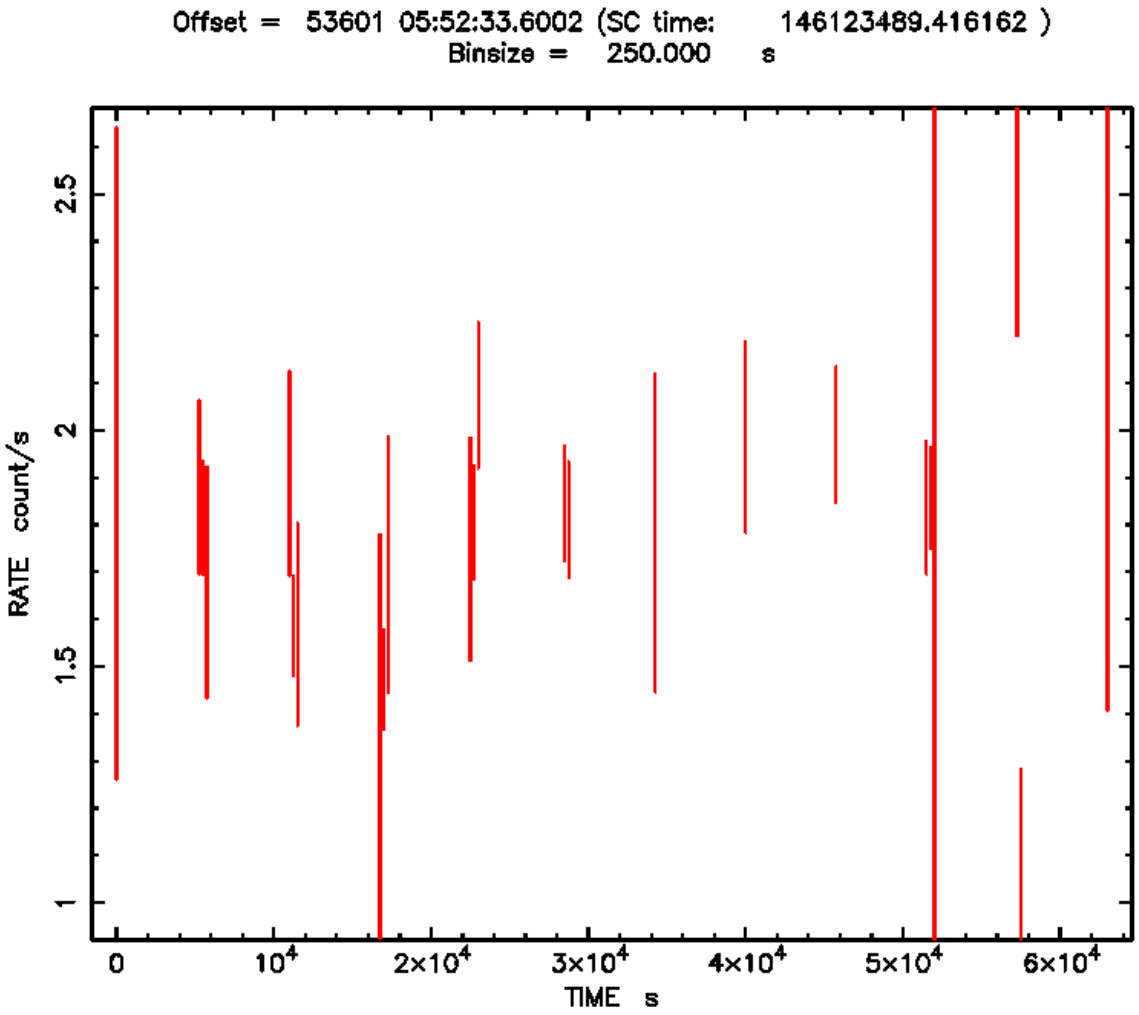


PC image

WT image

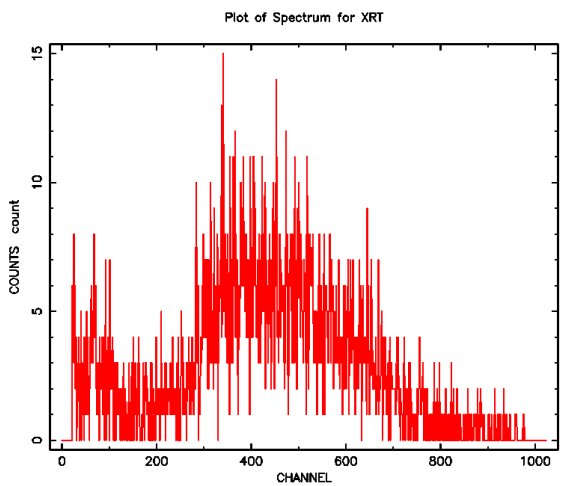


XSELECT light-curve



XSELECT - 3

- filter time cur – allows GTIs to be defined from the light-curve; can also do filter time file <filename> if there is a GTI file already or filter time scc (= spacecraft clock – i.e. time since start of observation).
- set binsize 250 – if you want time bins of 250s; type this BEFORE lc extraction!
- extract spectrum – uses all previous filtering (grades, regions, times)
- save spectrum – by default this will be a PI file, suitable for XSPEC and for use with the RMFs (providing no filtering with pha_cutoff was .



Spectrum within XSELECT

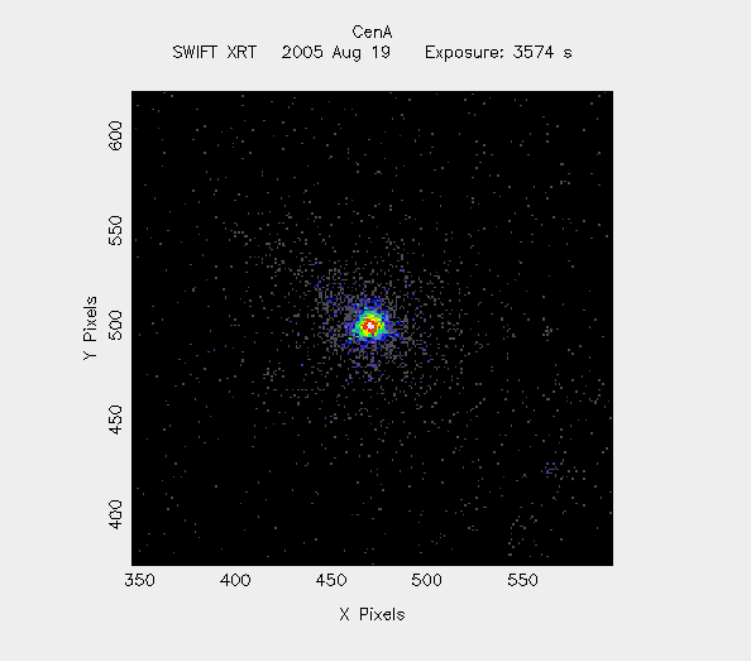
Needs to be read into XSPEC to do anything useful!

1 channel = 10 eV

Pile-up - 1

Although Swift has different modes designed for different count-rates, there can be problems with pile-up (generally in PC mode, but really bright objects can be piled-up in WT), where more than one photon hits a single (or adjacent) pixel before the charge has been read out. Where this is the case, an annular region should be used to extract the spectrum, rather than a circle, in order to exclude the core of the PSF. A quick way to estimate the radius which should be excluded is to run the PSF command in XIMAGE:

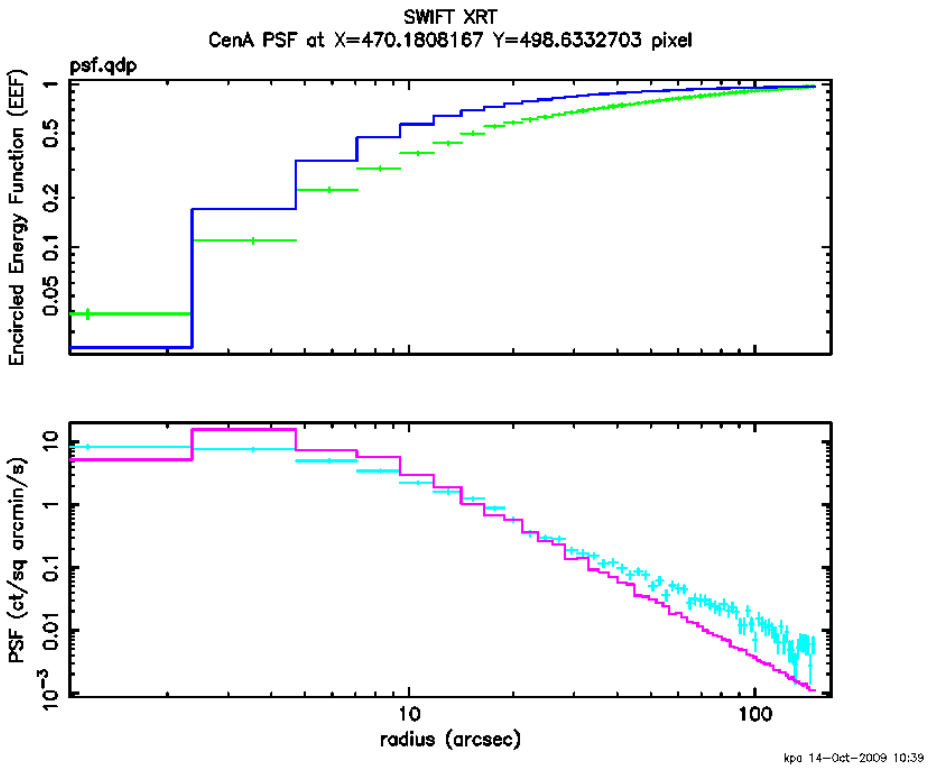
- > ximage
- > read sw[obsid]xpcw3posk.img
- > cpd /xtk
- > disp
- > psf/cur
  - click in centre and as far out as you want to determine the PSF



Pile-up - 2

This then takes you into QDP, with the PSF profile, which can be fitted with the King function ("model king") out in the wings – away from pile-up. The parameters measured from in-flight calibration are:

$$r_c \sim 5.8;\ \beta \sim 1.55 \quad \text{where } PSF(r) = [1 + (r/r_c)^2]^{-\beta}$$



Either use the command

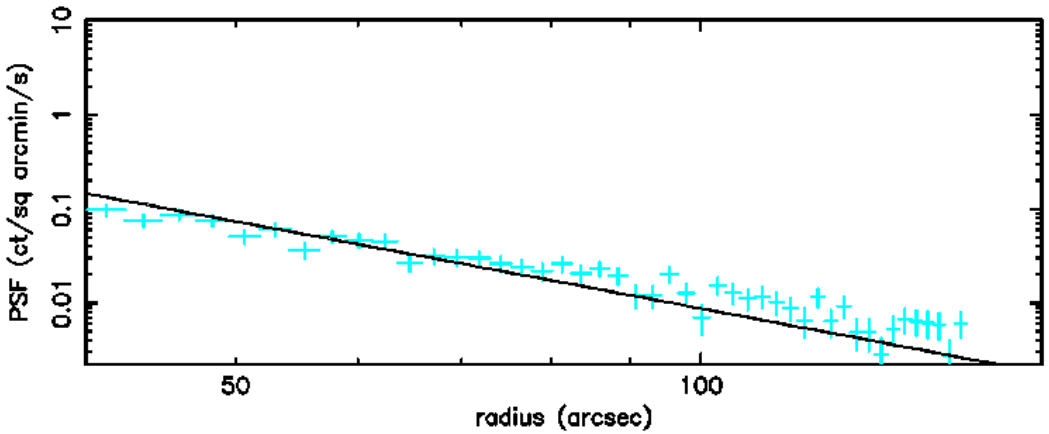
col off 1 2 3 4 6

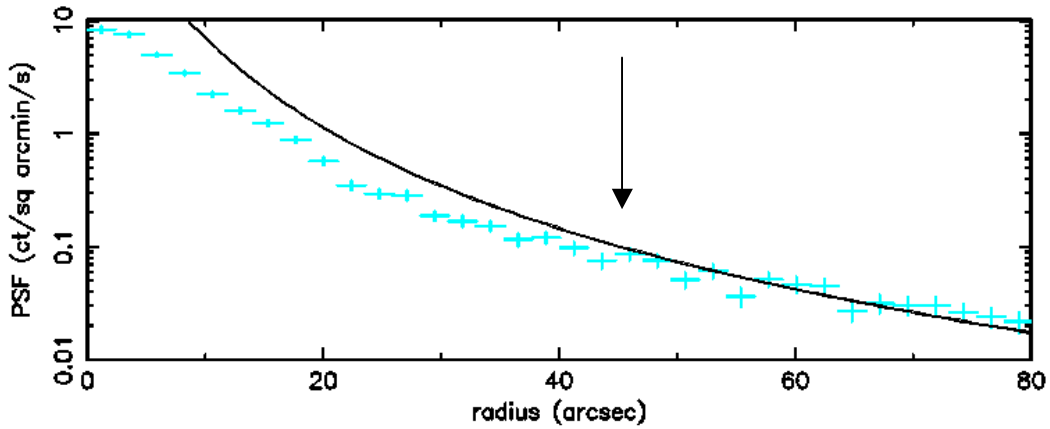
or

fit on 5

The cyan-coloured data points in the lower panel are the only ones we're interested in.

Pile-up - 3

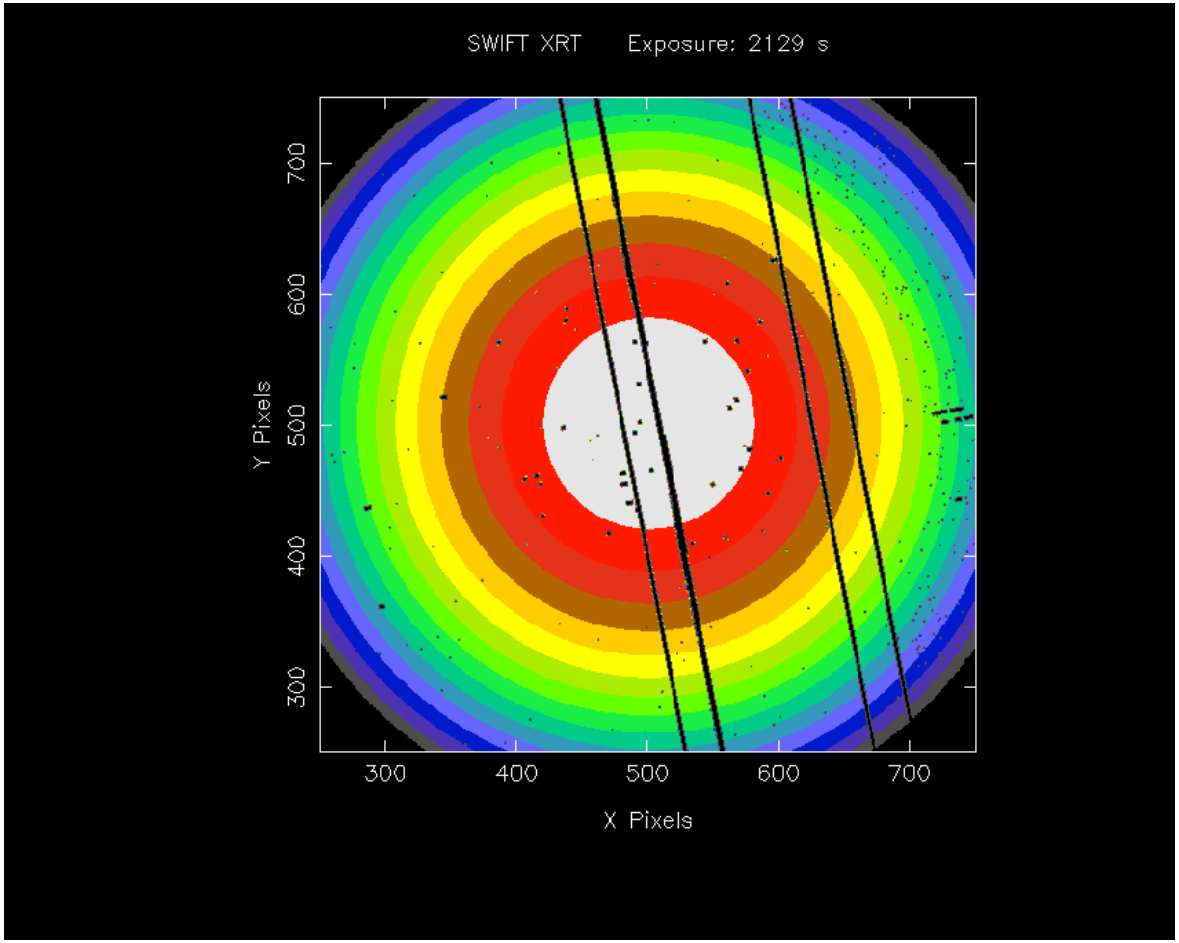




The black line shows the King profile fitted to the wings of the PSF where pile-up is negligible. Extrapolating down to smaller radii shows where the model and data start to diverge; this is the radius which should be excluded when extracting spectra/light-curves.

See also
http://www.swift.ac.uk/pileup.shtml

Exposure Maps - 1



Positioning a source (even partly) over the bad columns leads to a loss of flux. In order to correct for this, an exposure map needs to be generated for each orbit of interest.

This figure shows an example exposure map for PC mode data, demonstrating the bad columns and pixels in the field of view.

Exposure Maps - 2

When fitting a spectrum, an exposure map is required for whichever time interval is being considered, otherwise the calculated flux will be incorrect. The first step is to extract an eventlist for the relevant period of time. If you are using an entire Obs ID, then the default *cl.evt file is fine. Such an exposure map is automatically created if xrtpipeline was run with the command createexpomap=yes.

If a shorter interval is required:

Use filter time cursor to pick the orbit(s) of interest within XSELECT. Then a relevant event list needs to be produced:

> extract event copyall=yes

Note that the copyall=yes part is important; it keeps the "badpix" extension, which is needed to run the exposure map command.

If you wish to continue using the full dataset within this XSELECT session, make sure you say NO when prompted to say whether this new event list should be used as the input data file; then type clear event.

Exposure Maps - 3

xrtexpomap is the name of the Swift FTool which generates exposure maps. For each separate event file, this should be run as:

```
> xrtexpomap attfile=[obsid]/auxil/sw[obsid]sat.fits
hdfile=[obsid]/xrt/hk/sw[obsid]xhd.hk infile=orbit1.evt stemout=orbit1  outdir=./
```

This will make an exposure map called orbit1_ex.img which should then be used when creating the ARF.

Exposure Maps - 3

If you wish to combine more than one exposure map (for example if you want a single spectrum formed from more than one Obs ID – each has separate sat.fits and xhd.hk files, so they need to be considered one by one), they can be added within XIMAGE. Let's say we have a spectrum from datasets 001, 002 and 003. From these event lists, we have produced exposure maps 001_ex.img, 002_ex.img and 003_ex.img. Then:

```
> ximage
> read 001_ex.img
> read 002_ex.img
> sum
> save
> read 003_ex.img
> sum
> save
> write/fits 001-003_ex.img
```

Because the vignetting keyword is not copied by XIMAGE, the user should type (for e.g.)

```
> fparkey T 001-003_ex.img+0 VIGNAPP add=yes
```

Ancillary Response Files - 1

Ancillary Response Files (ARFs) can be built using the tool xrtmkarf. Note that Swift calibration is always on-going!

> xrtmkarf expofile=sw[obsid]xpcw3po_ex.img

Name of the input PHA FITS file [ ] : PC.pi

PSF correction active?(yes/no) [ ] : yes NB. "no" for extended sources.

Name of the output ARF FITS file [ ] : PC_exp.arf

Source X coordinate (SKY for PC and WT modes, DET for PD mode): [ ] : -1

Source Y coordinate (SKY for PC and WT modes, DET for PD mode): [ ] : -1

-1 takes the centre of the extraction region (information stored in the header of the spectrum) as the position of the source.

The relevant RMF will be given in the output written to the screen. These are located in the CALDB, wherever it is installed on your system.

Ancillary Response Files - 2

For both pile-up and exposure map corrections, the ratio between the corrected and uncorrected ARFs needs to be determined to calculate the factor by which light-curve data points should be multiplied, since only the spectra can make use of the corrected ARF directly.
To do this, 2 ARFs need to be built, with and without the expofile/psf=yes options and the predicted fluxes compared.

i.e., as well as the example on the previous page, also run:

```
> xrtmkarf
Name of the input PHA FITS file [ ] : PC.pi
PSF correction active?(yes/no) [ ] : no
Name of the output ARF FITS file [ ] : PC_no.arf
```

Then, the easiest method for determining the ratio is:

```
> fstatistic arf specresp -  (where arf = name of ARF)
```

Then compare the "max of selected column" for the ARF with/without the corrections. The ratio is the correction factor.

xrtlccorr

There is a useful task called xrtlccorr which calculates the bad column correction factor for every individual orbit and corrects the corresponding light-curves.

› xrtlccorr pcnframe=0 (or wtnframe=0 for WT data)

Name of the input region file or NONE to read region from lcfile: NONE

Name of the input Light Curve FITS file or NONE to read region from regionfile: PC.lc

Name of the Corrected Light Curve : PC_corr.lc

Name of the output file: pc.corr

Name of the input Attitude FITS file : sw[obsid]sat.fits

Name of the output Instrument Map File : DEFAULT

Name of the input Event FITS file : sw[obsid]*cl.evt

Name of the input Housekeeping Header Packets FITS file : sw[obsid]xhd.hk

The pc.corr file has a correction factor for every orbit of data (or every 10s if you don't include pcnframe=0). The input file can be for a single orbit or the whole observation – either will be corrected. The output file will always list the correction factors for every orbit in the cl.evt file, though.

lcmath

The FTool lcmath should be used to perform background subtraction on the light-curve files:

> lcmath

Name of input FITS file [ ] PC_corr.lc

Name of background FITS file [ ] PCback.lc

Name of output FITS file [ ] PC_corr_sub.lc

Scaling factor for input [ ] 1.

Scaling factor for background [] 0.25

Add instead of subtract? [ ] no

The background scaling factor is simply the ratio of the area of the source extraction region to that of the background region. So, if the source region has a radius of 30 pixels and the background 60 pixels, the background scaling factor would be $\pi 30^2/\pi 60^2 = 0.25$

flx2xsp - 1

There's a handy little FTool called flx2xsp which takes an ASCII file and converts it to a pha/rsp combination which works in XSPEC, thus allowing all the “spectral” models (power-law, broken power-law etc) to be fitted.

For each light-curve bin, the input file has to be in the format:

tstart tstop CR*(tstop-tstart) CR_Err*(tstop-tstart)

i.e., if you've extracted a light-curve in XSELECT, with 5 s binning, for example, the light-curve file itself would look something like this (NB Should background subtract using lcmath; may also need to correct for pile-up/bad columns…):

!tstart	CR	CR_Err
0	2.57969999	1.15367699
5	2.06376004	1.03188002
10	1.03188002	0.729649305
15	3.6115799	1.36504889

...

This is particularly useful for objects like GRBs where the brightness varies a lot over time, so fixed time bins are not really appropriate.

flx2xsp - 2

If the start time of this observation is 4745.36 s after the trigger time, this needs to be converted into a file like:

4745.36 4750.36 12.9 5.77

4750.36 4755.36 10.32 5.16

4755.36 4760.36 5.16 3.65

If you were to save this in a file called pc1.txt, then you'd run:

> flx2xsp pc1.txt pc1.pha pc1.rsp

pc1.pha can then be read into grppha and XSPEC as though it were a spectrum. (See later for XSPEC introduction.)

Note that flx2xsp has to be done on each (bright) orbit individually; the software does not cope with the long orbit gaps.

flx2xsp - 3

At later times, you might only want 1 bin per orbit (whatever the number of counts in there, as long as it's a significant detection), or even 1 bin spanning a number of orbits, when it's even fainter.

In this case, you can still use flx2xsp. For example, say we have 2 orbits spanning 1e4 to 3e4 seconds after the trigger, with an on-source exposure time of 10 ks. During this time, there may be 25 background-subtracted source counts. In that case, I would make a 1-line ASCII file:

10000 30000 50 10

where 10000 = tstart and 30000 = tstop

50 = (25/10000)*(30000-10000) ie the back-sub count rate * tstop-tstart

10 = (5/10000)*(30000-10000) where 5 is the error on the count rate.

This, of course, doesn't need to go through grppha after the flx2xsp step, since there's only 1 bin anyway.

Alternatively…

Although this presentation has tried to cover all the basics of how to extract XRT light-curves and spectra, there is a quick and easy way to "cheat"!

Light-curves (and hardness ratios) and time-averaged spectra are routinely created for every XRT-detected GRB and are available online:

http://www.swift.ac.uk/xrt_products

From these pages, there are also links to the Burst Analyser, which combines BAT and XRT data to create flux light-curves in different energy bands.

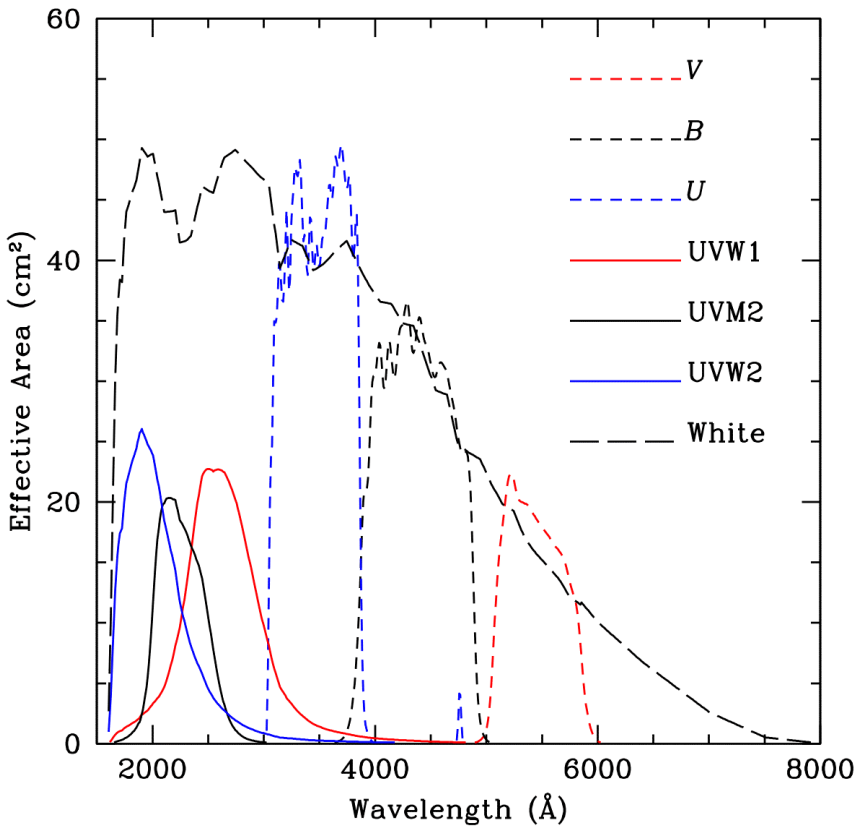
Similarly, if you want to create a light-curve or spectrum for a non-GRB source, go to

http://www.swift.ac.uk/user_objects

See Evans et al. (2009, MNRAS, 397, 1177), Evans et al. (2007, A&A, 469, 379) and Evans et al. (2010, submitted: arXiv:1004.3208) for details. These products are completely suitable for scientific analysis and have been used in many papers.

Swift - UVOT

UVOT filters



Filter	Central Wavelength (Å)
v	5468
b	4392
u	3465
uvw1	2600
uvm2	2246
uvw2	1928

UVOT data

UVOT data can be collected in Image, Event or Image&Event mode. Image mode provides a 2-D sky map with a start and stop time for the exposure as a whole, but not for the individual photons; Event mode does have time-tagged photons.

Early GRB observations will have both Image and Event mode data, but later observations will generally just be in Image mode.

Aspect Correction - 1

Most UVOT analysis for GRBs can be performed directly from the UVOT images available in the archive. You must check that the data are aspect corrected first, otherwise not all of the tools will work.

- fkeyprint [obsid]/uvot/image/sw[obsid]uuu_sk.img.gz ASPCORR

If ASPCORR = DIRECT, the correction has been applied. If this is not the case, uvotskycorr needs to be run TWICE (first to ID the sources and then to use that output correction file to WCS-correct the image). The information about the star catalogue file required can be obtained from http://heasarc.gsfc.nasa.gov/docs/swift/analysis/threads/uvot_thread_aspcorr.html

Aspect Correction - 2

First, calculate the aspect solution:

> uvotskycorr catspec=usnob1.spec
What to do (ID|SKY): ID
Input SKY image file name(s): [obsid]/uvot/image/sw[obsid]uuu_sk.img.gz
Input corrections file: NONE
Input attitude file: [obsid]/auxil/sw[obsid]sat.fits.gz
Output file name: ucorr.fits

Then apply this solution to the file:

> uvotskycorr catspec=usnob1.spec
What to do (ID|SKY): SKY
Input SKY image file name(s): [obsid]/uvot/image/sw[obsid]uuu_sk.img.gz
Input corrections file: ucorr.fits
Input attitude file: [obsid]/auxil/sw[obsid]sat.fits.gz
Output file name: NONE

Summing Images

Once images have been aspect corrected, they can be summed, to allow the detection of fainter sources.

Either the separate extensions within a given file can be summed:

```
>uvotimsum
Name of input image file: [obsid]/uvot/image/sw[obsid]uuu_sk.img.gz
Output file name: usum.fits
```

or if multiple Obs IDs are required (note the image must now be uncompressed):

```
>cp usum.fits usum_all.fits
>fappend 001sum.fits usum_all.fits
```

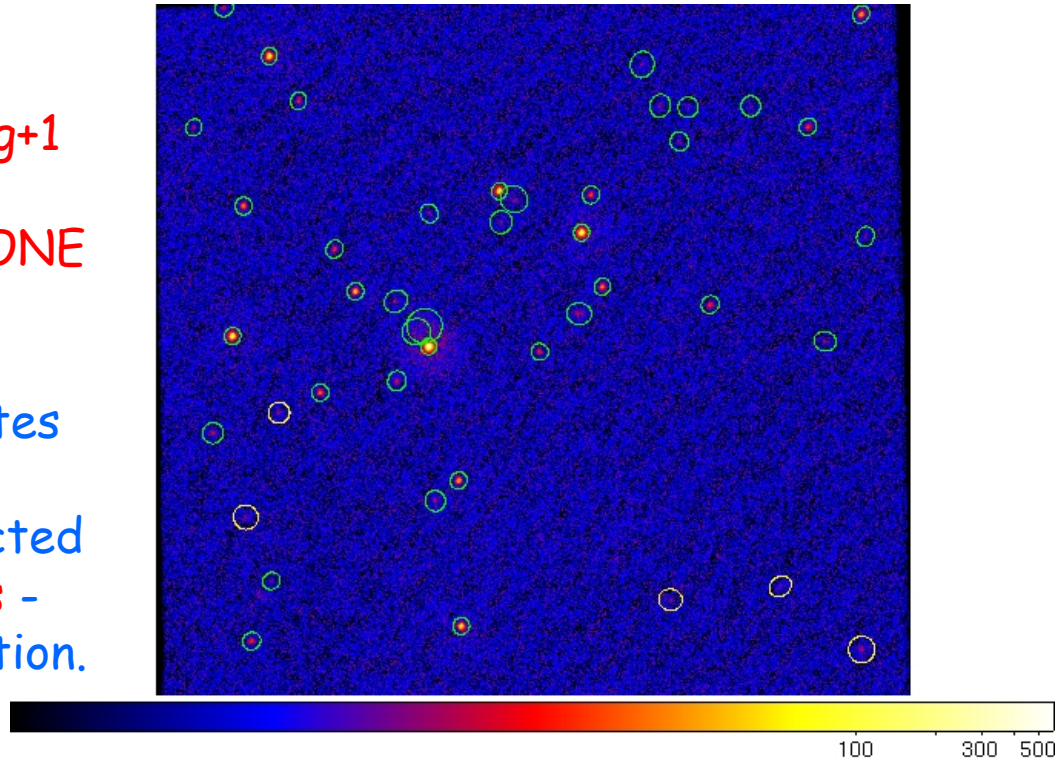
where 001sum.fits was created using uvotimsum on the segment 001 data. Then run uvotimsum on the completed usum_all.fits file. As many images as required can be added together in this manner.

Position Determination

The source detection routine uvotdetect detects all sources in a UVOT sky image, centroids and can optionally perform photometry. It works with individual or summed images but the FITS extension must be given; the image file can be used while zipped.

> uvotdetect plotsrc=yes
Input image file: sw[obsid]uuu_sk.img+1
Output file name: udetect.fits
Exposure map file name or NONE:NONE
Detection threshold (0.5:100):3

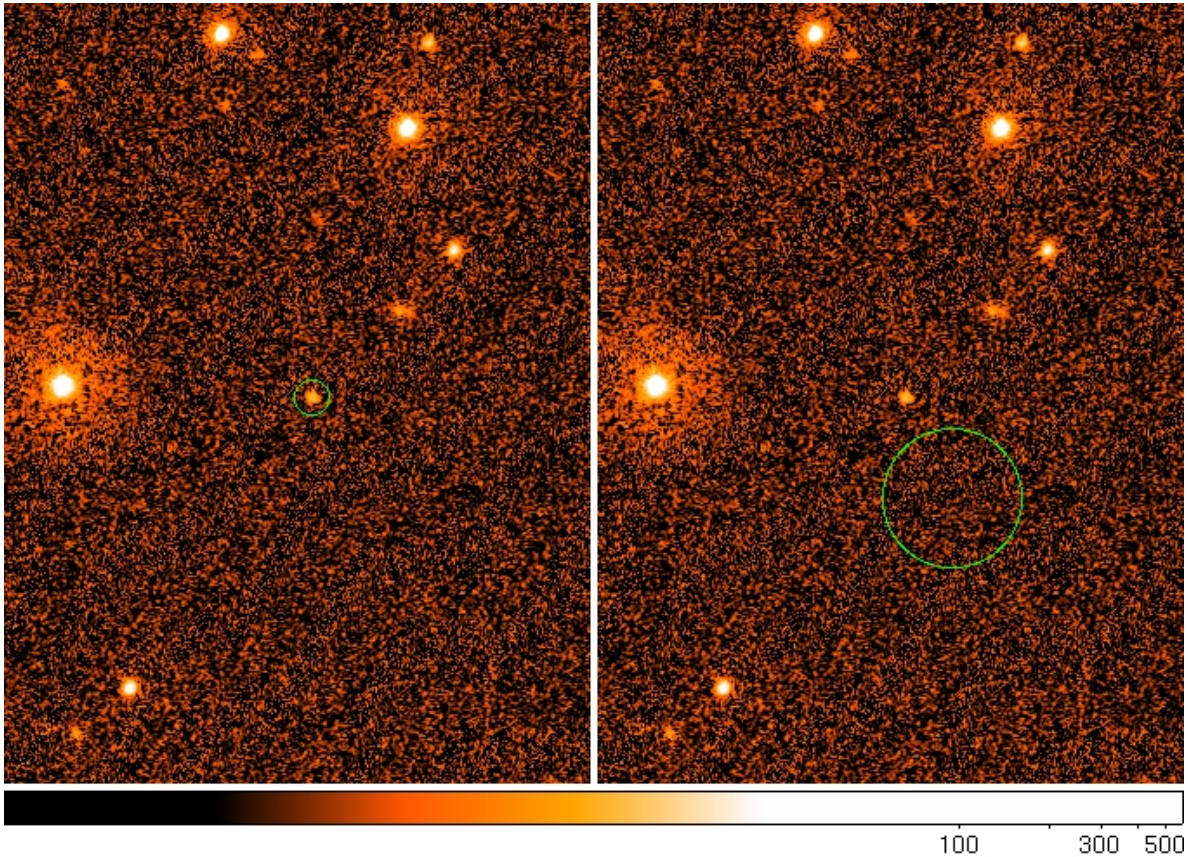
udetect.fits lists positions, count-rates and 1 σ errors for each source.
To perform photometry on the detected sources, set the option calibrate=yes - this includes coincidence loss correction.



uvotsource -1

uvotsource is the tool used to perform aperture photometry on a single source in a UVOT sky image.

It is best to use a circle with a radius of 5 arcsec for the source, since this was how the calibration was done. It is preferable to define a background region several times larger - this can either be a single large region or a combination of small ones.



uvotsource -2

> uvotsource
Input image file and extension[ ]:[obsid]/uvot/image/sw[obsid]uuu_sk.img.gz+1
Source region file[ ]:source.reg
Background region file[ ]:back.reg
Background threshold (1:100) [ ]:3
FITS table of values[ ]: uphotometry.fits

Results are then written both to the screen and the output FITS file:
detection significance,
coincidence-corrected count rate, magnitude and flux density
in erg cm^{-2} s^{-1} $Å^{-1}$ and in mJy with 1-sigma error
(or "Background threshold"-sigma upper limits if the source is undetected - 3 σ in the above example).

uvotsource output

```
uvotsource: running uvotinteg
uvotsource: Source
          Position: RA = 04h 00m 42.59s, Dec = -
55d 57m 20.0s (J2000)
          Position: RA = 60.17746, Dec = -55.95556
(J2000)
          Exposure: 245.85 s
            Filter: U
       Significance: 11.5 sigma (stat)
   Background-limit: 3.0 sigma (stat)
uvotsource: Magnitude [mag]
            Source: 18.50 +/- 0.10 (stat) +/- 0.02
(sys)
        Background: 23.83 arcsec^-2
   Background-limit: 20.36
  Coincidence-limit: 11.91
uvotsource: Flux density [erg/s/cm^2/A at 3501 A]
            Source: 1.41 +/- 0.13 (stat) +/- 0.02 (sys)
x 10^-16
        Background: 1.04 +/- 0.02 (stat) +/- 0.02
(sys) x 10^-18 arcsec^-2
   Background-limit: 2.55 x 10^-17
  Coincidence-limit: 6.07 x 10^-14
uvotsource: Corrected rate [count/s]
            Source: 0.866 +/- 0.078 (stat)
        Background: 0.0064 arcsec^-2
   Background-limit: 0.156
  Coincidence-limit: 372.234
uvotsource: Raw rate [count/s]
            Source: 0.856 +/- 0.075 (stat)
        Background: 0.0064 arcsec^-2
   Background-limit: 0.155
  Coincidence-limit: 90.644
uvotsource: Flux density [mJy at 8.563 x 10^14 Hz]
            Source: 5.77 +/- 0.52 (stat) +/- 0.09 (sys) x
10^-2
        Background: 4.26 +/- 0.10 (stat) +/- 0.07 (sys) x
10^-4 arcsec^-2
   Background-limit: 1.04 x 10^-2
  Coincidence-limit: 2.48 x 10^1
uvotsource: running ftcreate
uvotsource: running fthedit
uvotsource: running fthedit
```

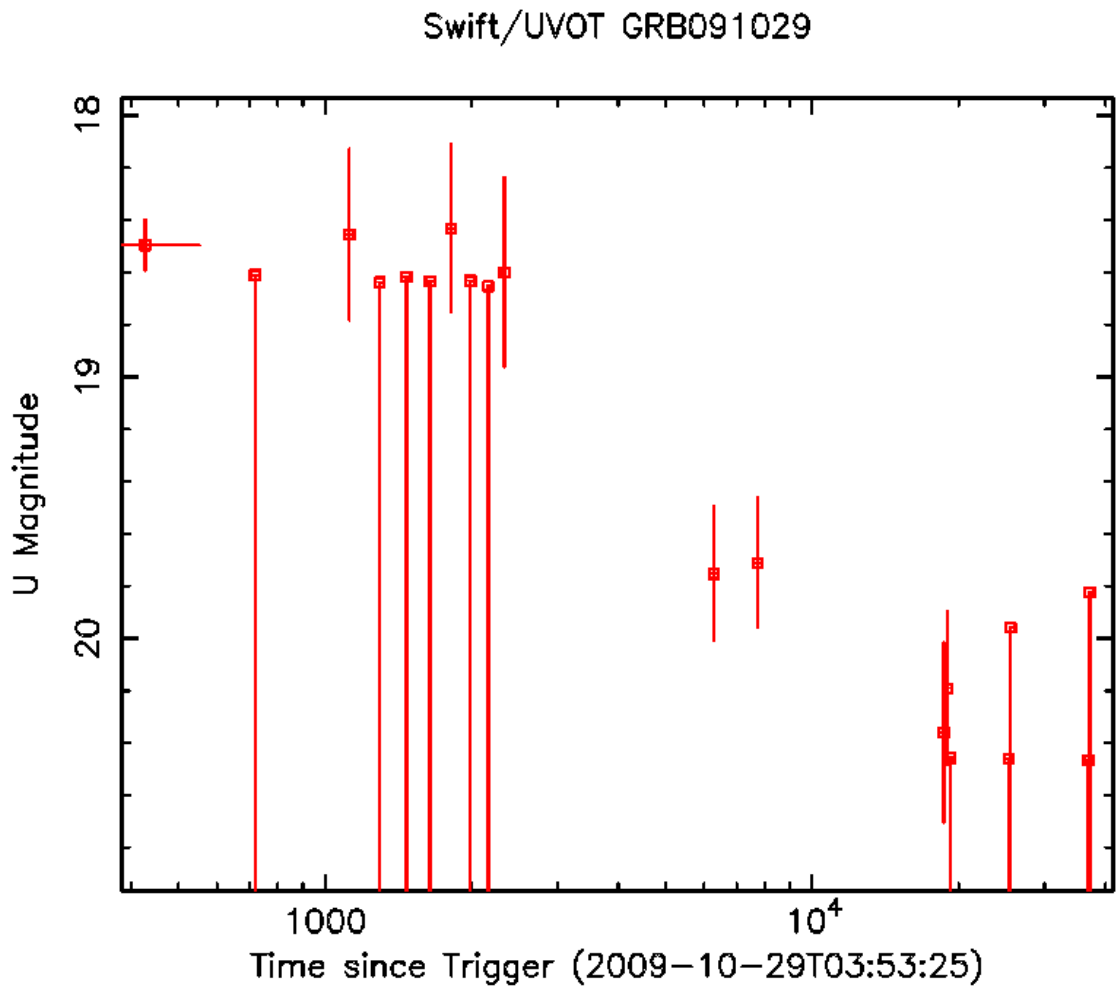
uvotmaghist - 1

The uvotmaghist tool uses uvotsource on all the given aspect-corrected extensions of an image to create a light curve.

```
> uvotmaghist
Name of input image file[ ]: [obsid]/uvot/image/sw[obsid]uuu_sk.img.gz
Name for output magnitude history[ ]: umaghist.fits
Name for output magnitude history plot or NONE[ ]: umaghist.gif
Name of zero points file or CALDB[ ]: CALDB
Name of coincidence loss file or CALDB[ ]: CALDB
Source region file or NONE[ ]: source.reg
Background region file or NONE[ ]: back.reg
```

The gif output file is a plot of magnitudes with time, though this can be changed using the plotcol command. One point will be plotted for each extension of the image. If any of these are non-detections (at the 3-sigma level, by default), the point will be plotted with just a negative error bar. The FITS file contains all the results as well.

uvotmaghist - 2



uvot2pha

To create PHA files for input to XSPEC from the same UVOT images used for the uvotmaghist task, first locate the latest canned response matrices for the filters you require from the CALDB. These are named swu‹filter›_dateversion.rsp, where dateversion will be something like 20041120v104. Run uvot2pha, adding the extension number to the input filename.

```
>uvot2pha
Input (img or evt) filename[ ]: [obsid]/uvot/image/sw[obsid]uuu_sk.img.gz+1
Source region file name[ ]: source.reg
Background region file name[ ]: back.reg
Output source PHA filename[ ]: usource.pha
Output background PHA filename[ ]: uback.pha
Response file name[ ]: swuuu_20041120v104.rsp
```

Event mode - 1

If event mode data have been collected, they will appear as *uf.evt files in the archive. There are 2 steps needed to clean these data: convert the raw coordinate positions into detector and sky coordinates, then screen out the hot pixels:

```
>coordinator eventfile=[obsid]/uvot/event/sw[obsid]uuuw1po_uf.evt
eventext=EVENTS teldef=CALDB attfile=[obsid]/auxil/sw[obsid]sat.fits.gz
aberration=n randomize=y seed=836 ra=www.xxx dec=yyy.zzz
```

where the w1 part of the eventlist name signifies the window size. (Note that you may get w1w1 in the filename if the filter used was uvw1!)
The *uf.evt file should be unzipped before this command is run.
The seed is just a random number to start off the task.
The RA/Dec. values (use decimal format) will be used by software such as ds9 to centre the image.
NB coordinator overwrites the input file.

Event mode - 2

```
>uvotscreen infile=[obsid]/uvot/event/sw[obsid]uuuw1po_uf.evt
attorbfile=[obsid]/auxil/sw[obsid]sao.fits.gz outfile=sw[obsid]uuuw1po_cl.evt
badpixfile=caldb aoexpr="ANG_DIST< 100. && ELV > 10. && SAA == 0"
evexpr="QUALITY==0"
```

This cleaned eventlist can then be read into XSELECT and light-curves extracted just as for the XRT. The minimum time resolution for UVOT event mode data is 0.011033 s. It's a good idea to use a multiple of this number when extracting light-curves.
Alternatively, uvotevtlc is a task specifically designed for UVOT event mode light-curves:

```
>uvotevtlc timebinalg=u timedel=10
Input event filename [ ]: sw[obsid]uuuw1po_cl.evt
Output light curve filename [ ]: u.lc
Source region filename [ ]: src.reg
Background region filename [ ]: back.reg
```

"timebinalg=g gtifile=gti" is the alternative, where gti is the relevant GTI file.

XMM-Newton - EPIC

Set up and processing

The Observation Data Files (ODFs) are needed and should be pointed to using the command

> setenv SAS_ODF <path to ODFs>

Then a Current Calibration File (CCF) needs to be build:

> cifbuild withccfpath=yes ccfpath=<path to cal. files> fullpath=yes

(which produces a file called ccf.cif) and pointed to:

> setenv SAS_CCF <path to ccf.cif>

In the ODF directory, run odfingest to create the SUM.SAS file which is a summary of all the files. MOS and PN data can then be processed.

NB. The FTools must be set up as well as the XMM software!

> epchain >& pn.log

> emchain >& mos.log (which processes both MOS1 and MOS2 data)

The processed eventlists are then called *PIEVLI000.FIT and *MIEVLI000.FIT

XMMSELECT

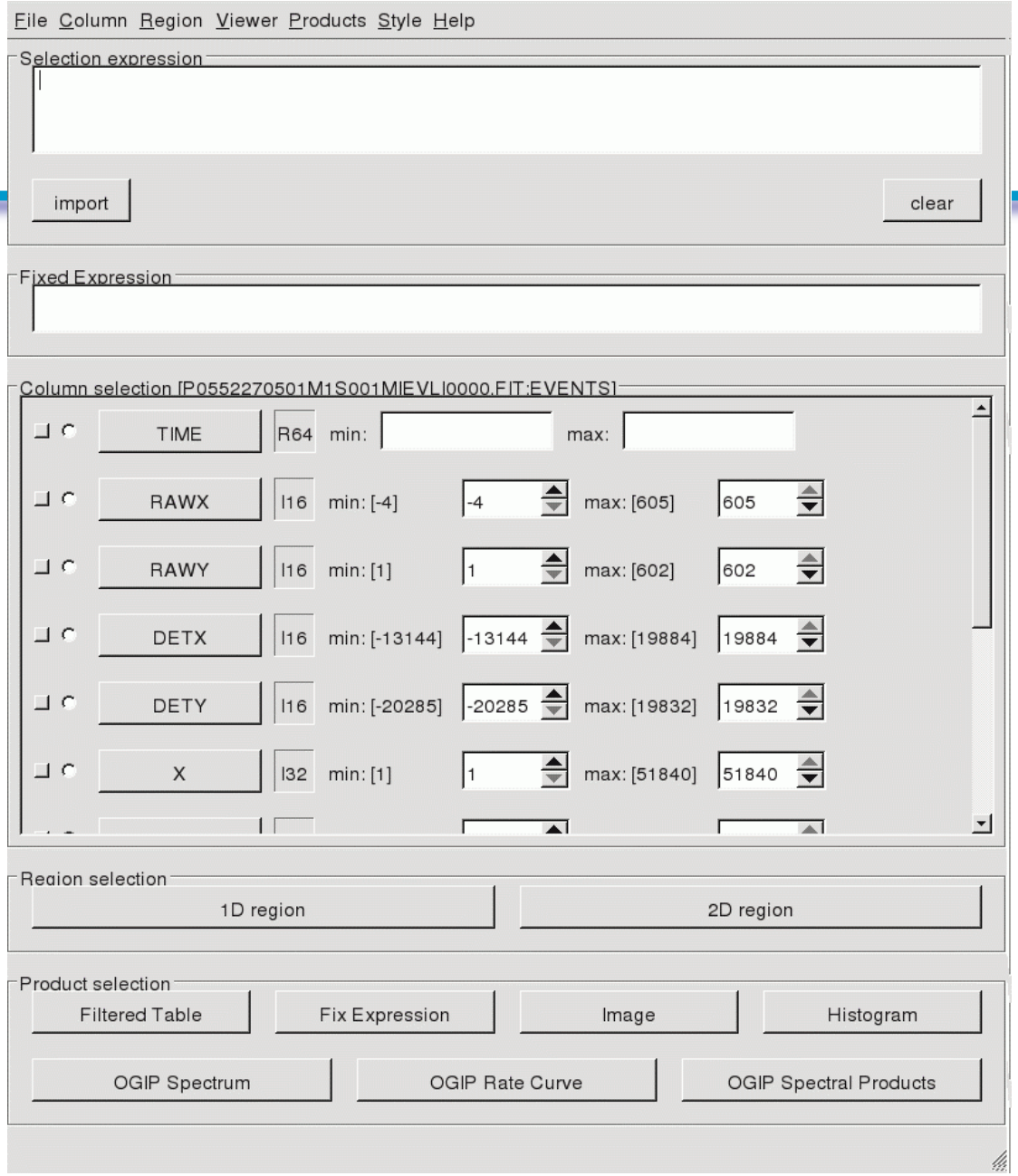
XMMSELECT is a Graphical User Interface (GUI) to EVSELECT (just as XSELECT is for extractor).

> xmmselect table=P[obsid]M1xxxxMIEVLI0000.FIT

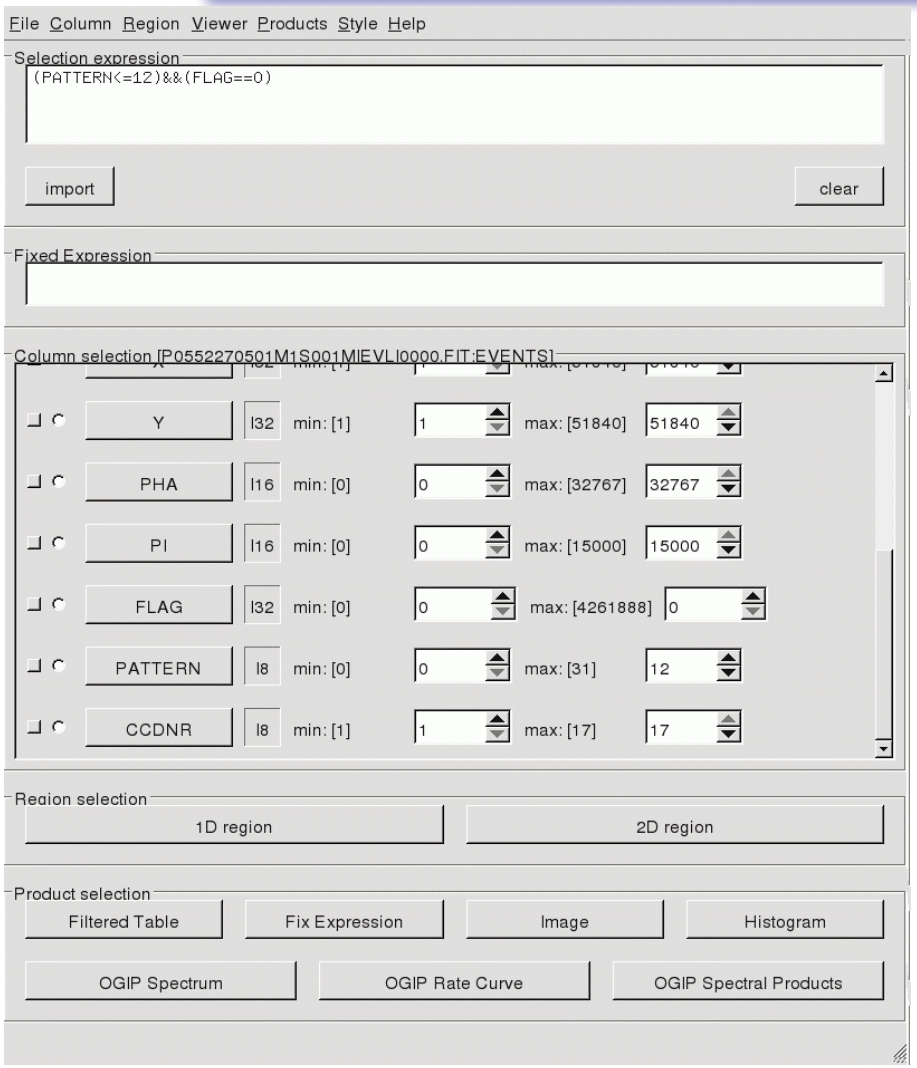
(You can click “NO” when asked about the subspace information. This just gives information about the default filtering already performed.)

This interface can then be used to extract images, light-curves and spectra. Note that the eventlists can also be used in XSELECT, but XMMSELECT has been especially designed for XMM data analysis.

MOS data are calibrated for PATTERN==0 (single pixel events only) or PATTERN<=12 (single, double, triple and quadruple events), pn for 0 or <=4 (singles and doubles).


File Column Region Viewer Products Style Help
Selection expression
import
clear
Fixed Expression
Column selection [P0552270501M1S001MIEVLI0000.FIT:EVENTS]
TIME R64 min: max:
RAWX I16 min: [-4] -4 max: [605] 605
RAWY I16 min: [1] 1 max: [602] 602
DETX I16 min: [-13144] -13144 max: [19884] 19884
DETY I16 min: [-20285] -20285 max: [19832] 19832
X I32 min: [1] 1 max: [51840] 51840
Region selection
1D region
2D region
Product selection
Filtered Table
Fix Expression
Image
Histogram
OGIP Spectrum
OGIP Rate Curve
OGIP Spectral Products

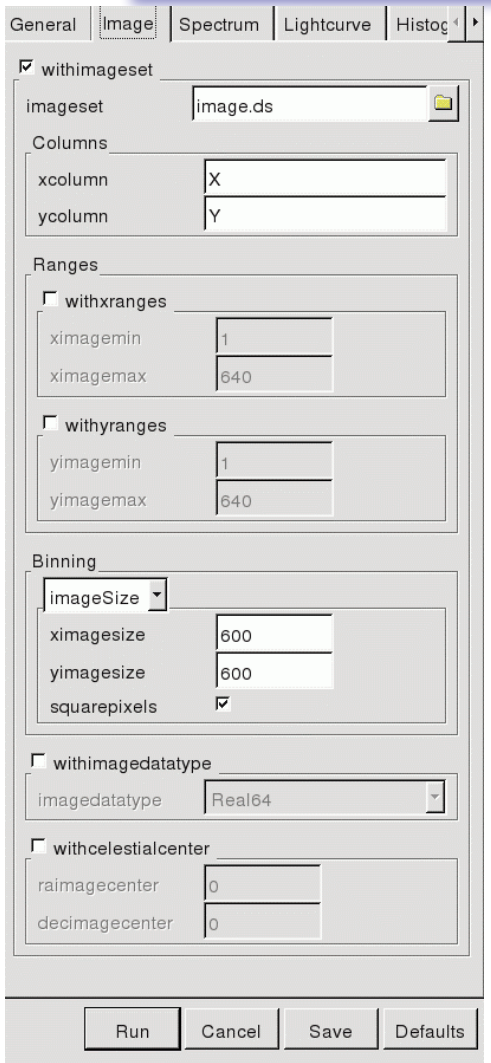
Setting the stage



Set the required PATTERN selection and FLAG==0 (bad events excluded) by typing the numbers into the white boxes and clicking the named button. This writes the selection expression to the top box. If energy filtering is required for a light-curve, use the PI boxes (measured in eV).

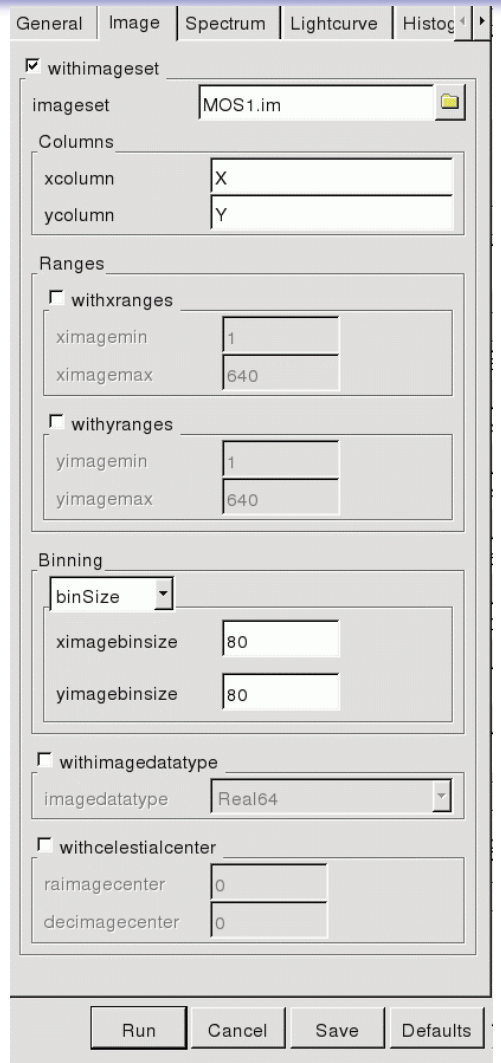
For a "normal" image, click the small square to the left of X and then the one to the left of Y, to define the x and y coordinate axes.

Extracting an image



Clicking on the Image button at the bottom opens a new window. Choosing the Image tab at the top brings up the window shown on the left. Change the imageset (i.e. the name) to something suitable and the binsize to 80 x 80.

Clicking on the Run button will open an image in DS9, within which source and background regions can be defined.



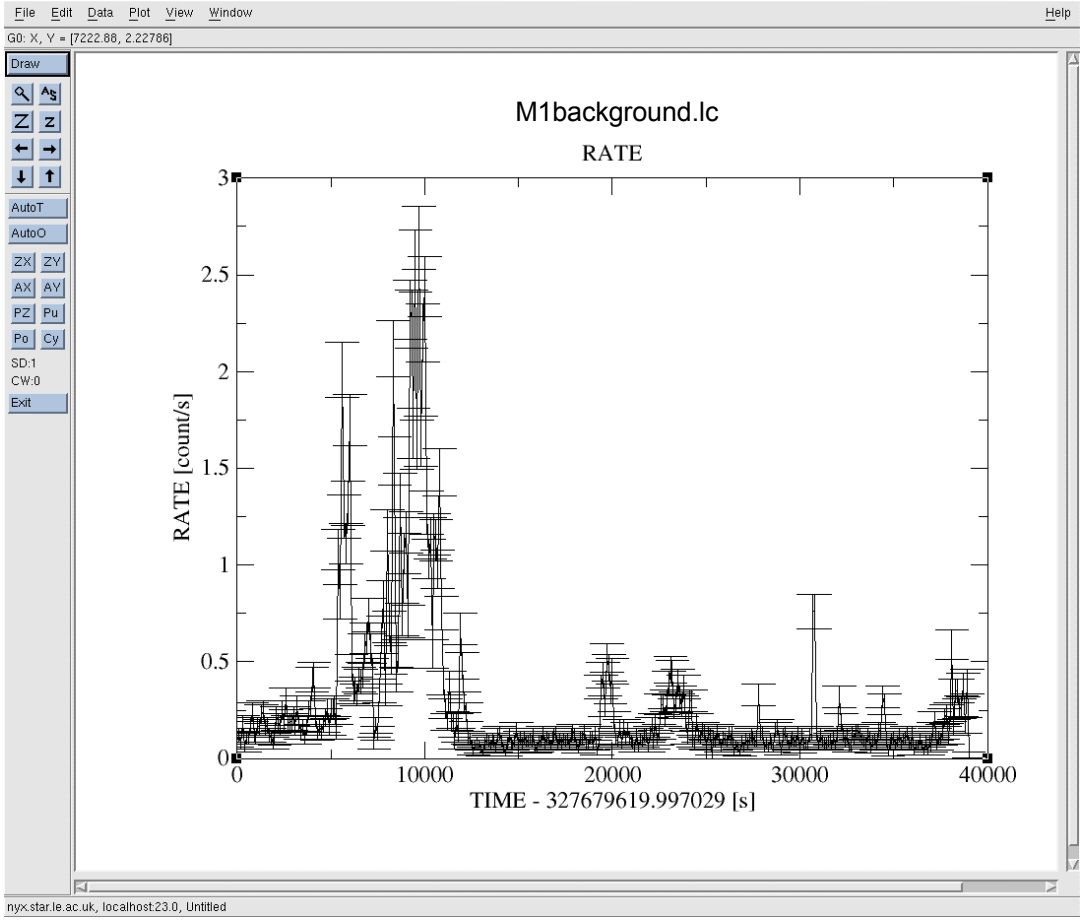
Checking for high background

The light-curve should be checked to identify periods of high background which can then be discarded. To do this, the selection expression at the top of the window should read
(PATTERN==0)&&(PI>=10000)&&#XMMEA_EM
(or #XMMEA_EP for pn data)
i.e. single events with energies above 10 keV.

To extract a light-curve, choose the round radio button at the left of the TIME row and click on OGIP Rate Curve to bring up another window as before. This time, choose the Lightcurve tab and choose a suitable time bin (25, 50 or 100 s would be sensible, depending on the exposure length).

Sometimes all the background will be low, but other times you'll see something like this:

Background light-curve



The XMM team recommend that a rate of 0.35 count s^{-1} for MOS or 1 count s^{-1} for pn is a suitable threshold at which to cut the data. To do this type the following command:

```
> tabgtigen
table=M1background.lc
gtiset=M1gti.fits
expression = ' RATE < 0.35 '
```

This would then be included in the selection expression when extracting a spectrum:

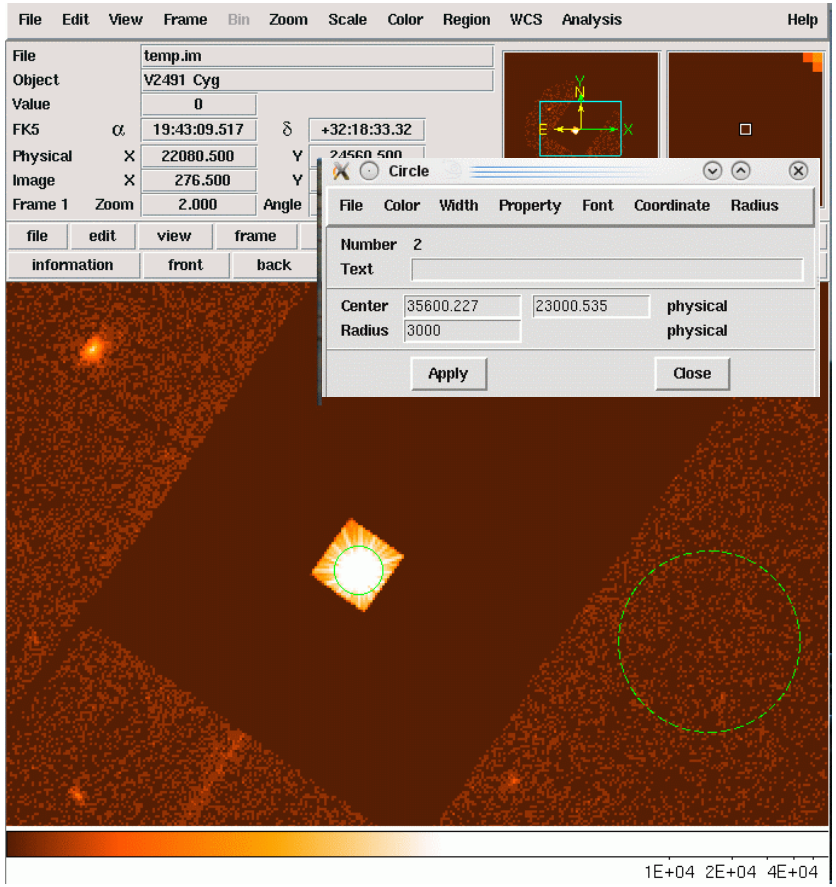
```
(PATTERN<=12)&&(FLAG==0)&&gti(M1gti.fits, TIME)
```

Spectra

XMMSELECT will extract source and background spectra and build the relevant ARFs and RMFs automatically, given the extraction regions required:

1. Extract an image and define source and background regions. If in small window mode, the background will need to be taken from a different chip. Double click on the background region, go to Property and choose Background.

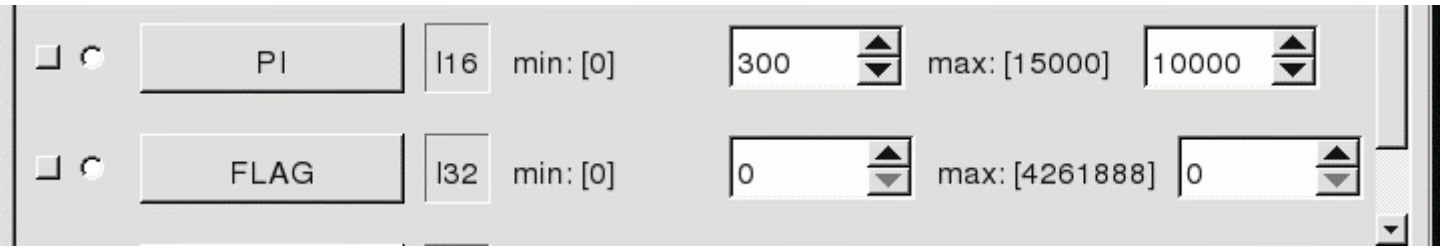
2. Click on the OGIP Spectral Products button. This will "optimise" the source region (if it's a circle) – use judgement!



Light-curves - 1

These have to be generated separately for the source and background – there's no equivalent of the Spectral Products button.

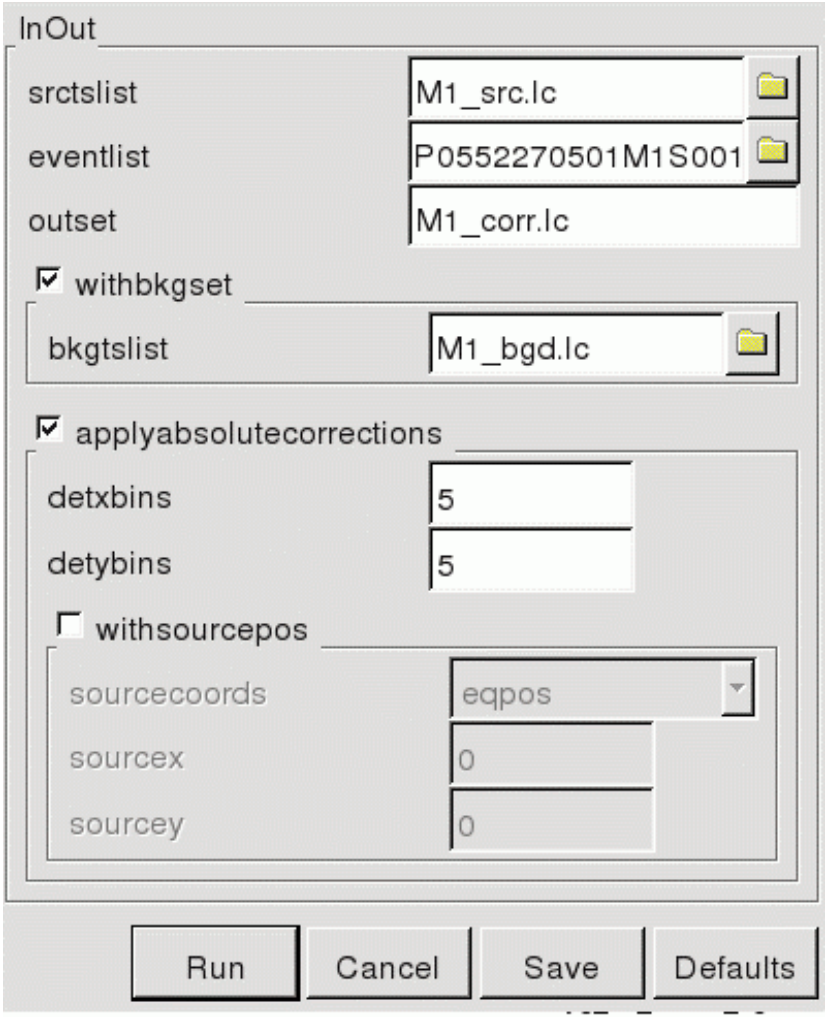
1. Extract an image, if not already done, and identify the source region required.
2. Click the 2D Region button to add the extraction region to the selection expression box.
3. Click the round radio button to the left of the TIME button.
4. If energy filtering is required (by default you get 0-15 keV), put the lower/upper limits in the PI white boxes and click the PI button.



Light-curves - 2

5. Click OGIP Rate Curve button and choose Lightcurve tab.
6. Change name and time bin size – and click Run!
7. Delete source region from selection expression and repeat for background region.
8. epiclccorr will subtract the background lc and make relevant corrections for dead time, GTIs etc. To run this using a GUI format, type epiclccorr –d &

Light-curves - 3



Fill in the names of the srctslist (= source light-curve), eventlist and outset (= required name of output file). Tick the withbkgset button and enter the name of the background light-curve. applyabsolutecorrections should also be ticked.

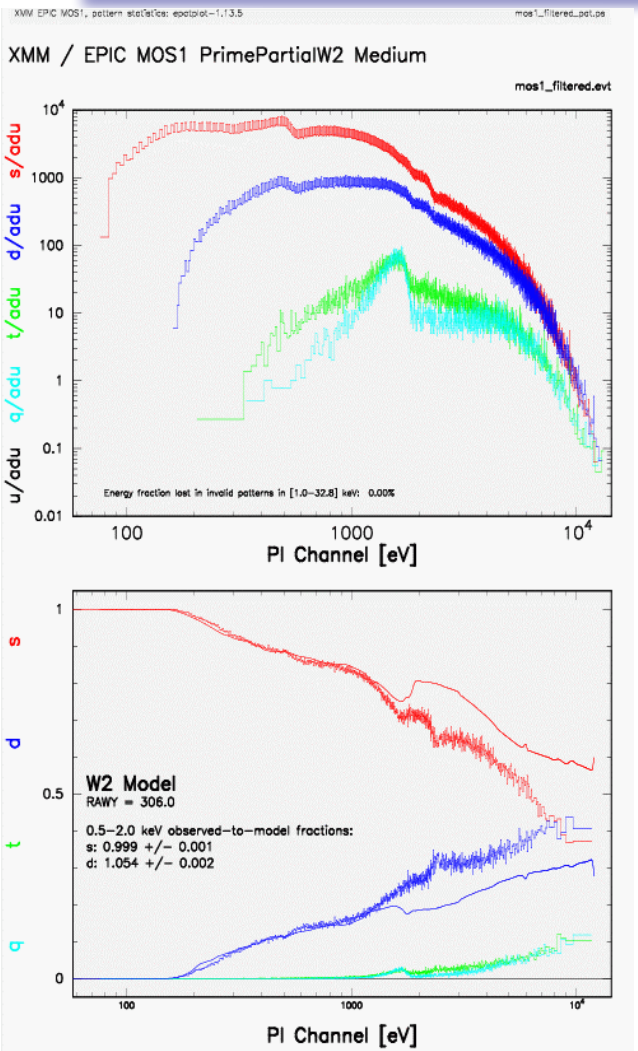
Pile-up - epatplot

epatplot is a script which allows you to estimate the level of pile-up in XMM-EPIC data. (A version will be available for Swift data in the future, as an alternative to the PSF-fitting in XIMAGE.)

1. Extract a filtered eventlist (Filtered Table button; then the General tab) for the source – in a circle to begin with, then annular regions to determine how much should be excluded. Do not filter on PATTERN (all are required for the eventlists).
2. Run epatplot on each of these eventlists:

   > epatplot set=mos1_filtered.evt

epatplot - 2



The bottom panel shows the actual (histograms of points) and expected (solid lines) pattern distribution. If pile-up is a problem, there will be too few single events (red histogram lower than solid line) and too many doubles (blue histogram below solid line). The excluded area should be increased until the histograms and lines match reasonably well. See http://xmm.esac.esa.int/sas/current/watchout/Evergreen_tips_and_tricks/eppatplot.shtml

This annulus should be used for both spectra and light-curves.

XMM-Newton - RGS

Processing

The method is very similar to that for EPIC data: point at the ODFs and CCFs and then:

› rgsproc >& log

This task automatically extracts spectra and generates response matrices for the prime source. The coordinates of the PROPOSAL source can be checked in the header of the source list (*SRCLI*). If these are correct (usually the case), then the extraction should be in the right place.

You can check the dispersion versus cross-dispersion and energy images to see where the regions lie. Although XMMSELECT can be used as previously shown, the command-line EVSELECT can also be run (works for EPIC as well), as follows:

Creating the dispersion plots

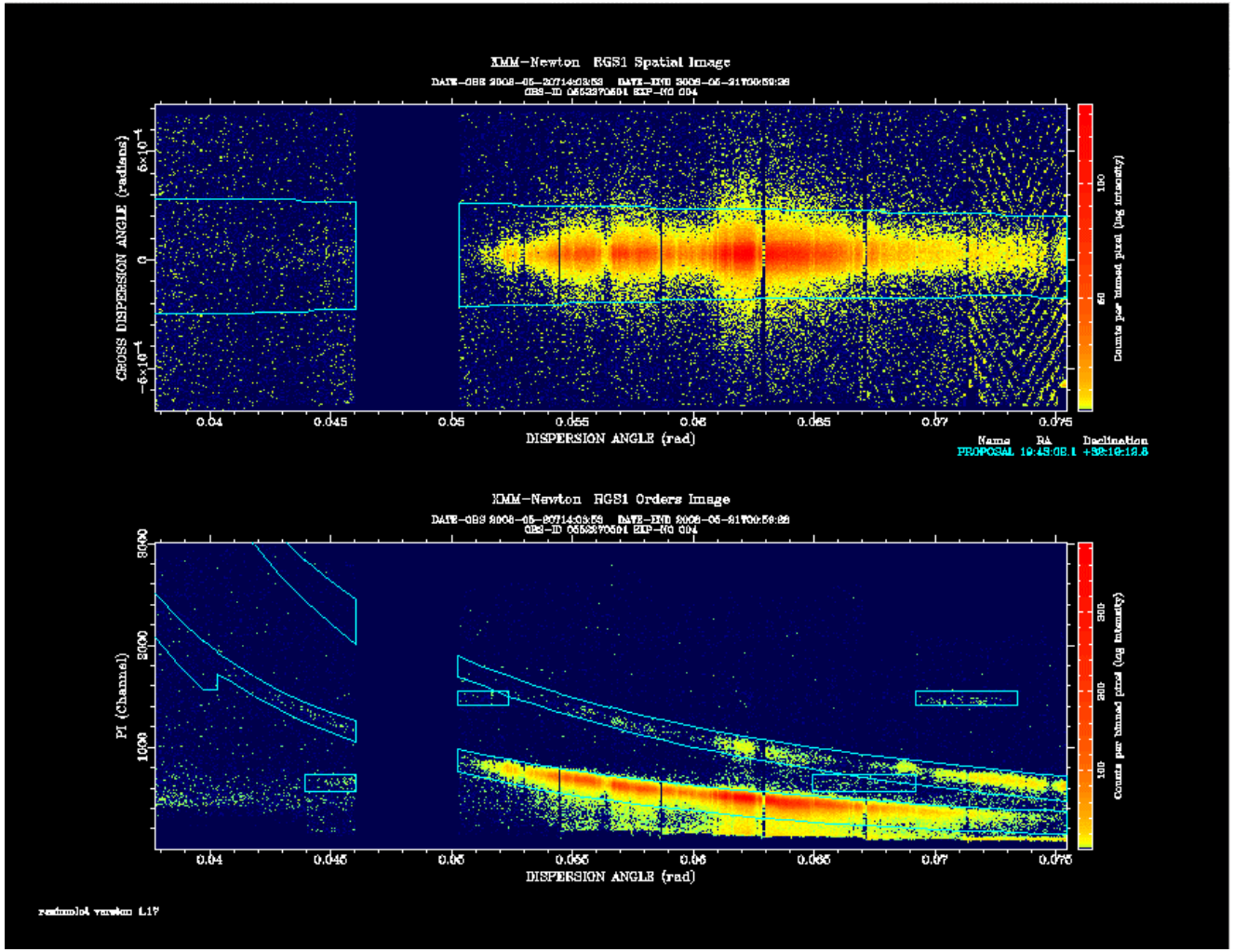
```
> evselect table='P[obsid]R1xxxxEVENLI0000.FIT:EVENTS'
imageset='spatial.fits' xcolumn='BETA_CORR' ycolumn='XDSP_CORR‘

> evselect table='P[obsid]R1xxxxEVENLI0000.FIT' imageset=‘pi.fits’
xcolumn='BETA_CORR' ycolumn='PI' yimagemin=0 yimagemax=3000
expression='REGION(P[obsid]R1xxxxSRCLI_0000.FIT:RGS1_SRC1_SPATIAL,
BETA_CORR,XDSP_CORR)‘

> rgsimplot endispset='pi.fits' spatialset='spatial.fits' srcidlist='1'
srclistset='P[obsid]R1xxxxSRCLI_0000.FIT' device=/xw
```

Likewise for the RGS2 data.

Dispersion Plots



Checking for high background

Again, this is similar to the EPIC process, though with a few additional components.

```
> evselect table=P[obsid] R1xxxxEVENLI0000.FIT timebinsize=100
rateset=RGS1_back.lc makeratecolumn=yes maketimecolumn=yes
expression='(CCDNR==9)&&
(REGION(P[obsid]R1xxxxSRCLI_0000.FIT:RGS1_BACKGROUND,BETA_CORR,
XDSP_CORR))'
```

Here, only events in CCD 9 – the one closest to the optical axis of the telescope, so most affected by background flares – are selected.
If flares are present, tabgtigen can be used as for the EPIC eventlists. Typically data with rates of >0.1-2 count s^{-1} should probably be excluded – decide by looking at the lc.
The data should then be reprocessed using this GTI file:

```
> rgsproc entrystage=3:filter auxgtitables=RGSgti.fits
```

XSPEC

Introduction and set-up

Although there are other spectral fitting packages available – ISIS, for example – XSPEC is widely used. It incorporates a large suite of models, ready to fit to your spectra, as well as allowing the user to write their own where something special is required.

In the past, many (if not most) people have used χ^2 statistics, but this method can bias results, so Cash-statistics are actually preferred. To use C-stat in XSPEC, the spectrum must have at least 1 count per bin. To ensure this, use grppha.

grppha also allows you to identify the relevant background, ARF and RMF, so they will be read into XSPEC automatically.

NB. This is NOT done for BAT data, since the spectra are grouped by channels. χ^2 statistics should be used, with no further binning within grppha.

grppha

```
> grppha
Please enter PHA filename [ ] WT.pi
Please enter output filename[ ] !WT.pi
GRPPHA[ ] bad 0-29
GRPPHA[ ] group min 1
GRPPHA[ ] chkey backfile WTback.pi
GRPPHA[ ] chkey ancrfile WT_exp.arf
GRPPHA[ ] chkey resp swxwt0to2s6_20070901v011.rmf
GRPPHA[] exit
 ... written the PHA data Extension
 ..... exiting, changes written to file : WT.pi
 ** grppha 3.0.1 completed successfully
```

(The ! preceding the file overwrites the same name.)

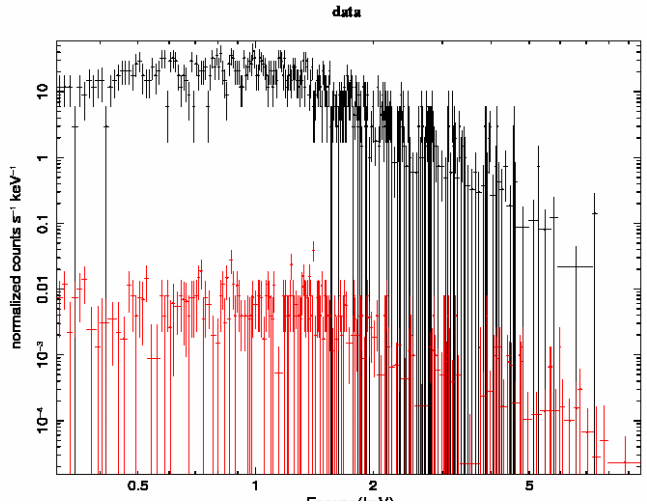
XSPEC - 1

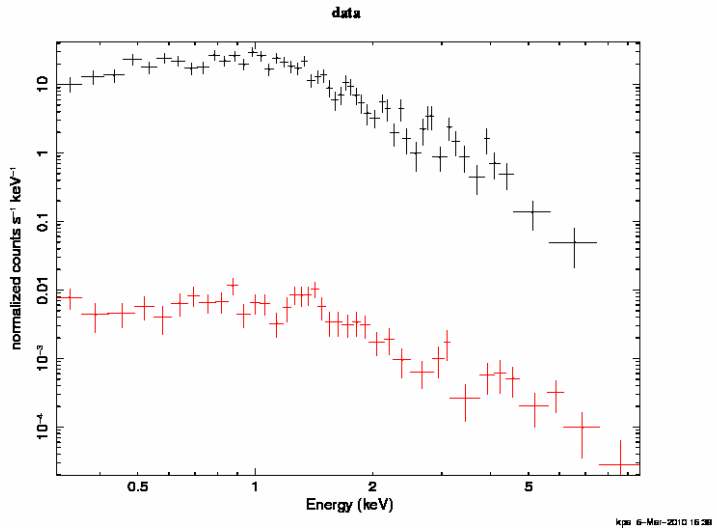
```
> xspec
 XSPEC version: 12.5.1n
      Build Date/Time: Tue Dec  8 15:18:05 2009
XSPEC12>data 1:1 WT.pi
***Warning: Detected response matrix energy bin value = 0 (or neg).
   XSPEC will instead use small finite value (response file will not be altered).
1 spectrum  in use
Spectral Data File: WT.pi  Spectrum 1
Net count rate (cts/s) for Spectrum:1  3.108e+01 +/- 9.665e-01 (99.4 % total)
 Assigned to Data Group 1 and Plot Group 1
  Noticed Channels:  1-566
  Telescope: SWIFT Instrument: XRT  Channel Type: PI
  Exposure Time: 33.65 sec
 Using Background File              WTback.pi
  Background Exposure Time: 33.65 sec
 Using Response (RMF) File   swxwt0to2s6_20070901v011.rmf for Source 1
 Using Auxiliary Response (ARF) File  WT_exp.arf
XSPEC12>data 2:2 PC.pi
```

The 1:1 and 2:2 means that the spectra can be fitted independently.

XSPEC - 2

```
XSPEC12>cpd /xw
XSPEC12>setpl energy
XSPEC12>ignore bad
ignore:   91 channels ignored from  source number 2
ignore:  308 channels ignored from  source number 1
XSPEC12>ignore **-0.3 10.0-**
   30 channels (1-30) ignored in spectrum #     1
   25 channels (542-566) ignored in spectrum #     1
XSPEC12>plot ld
XSPEC12>setplot rebin 20 5
XSPEC12>plot
```





Say we want to fit an absorbed power-law – fairly usual for an X-ray spectrum of a GRB:

XSPEC - 3

```
XSPEC12>mo con*po*wa
Input parameter value, delta, min, bot, top, and max values for ...
              1      0.01        0        0     1e+10     1e+10
1:data group 1::constant:factor>, -1
              1      0.01       -3       -2        9        10
2:data group 1::powerlaw:PhoIndex>
              1      0.01        0        0     1e+24     1e+24
3:data group 1::powerlaw:norm>1e-4, 1e-6
              1     0.001        0        0    100000     1e+06
4:data group 1::wabs:nH>

Input parameter value, delta, min, bot, top, and max values for ...
              1        -1        0        0     1e+10     1e+10
5:data group 2::constant:factor>, 0.01
              1      0.01       -3       -2        9        10
6:data group 2::powerlaw:PhoIndex>
         0.0001     1e-06        0        0     1e+24     1e+24
7:data group 2::powerlaw:norm>
              1     0.001        0        0    100000     1e+06
8:data group 2::wabs:nH>
```

XSPEC - 4

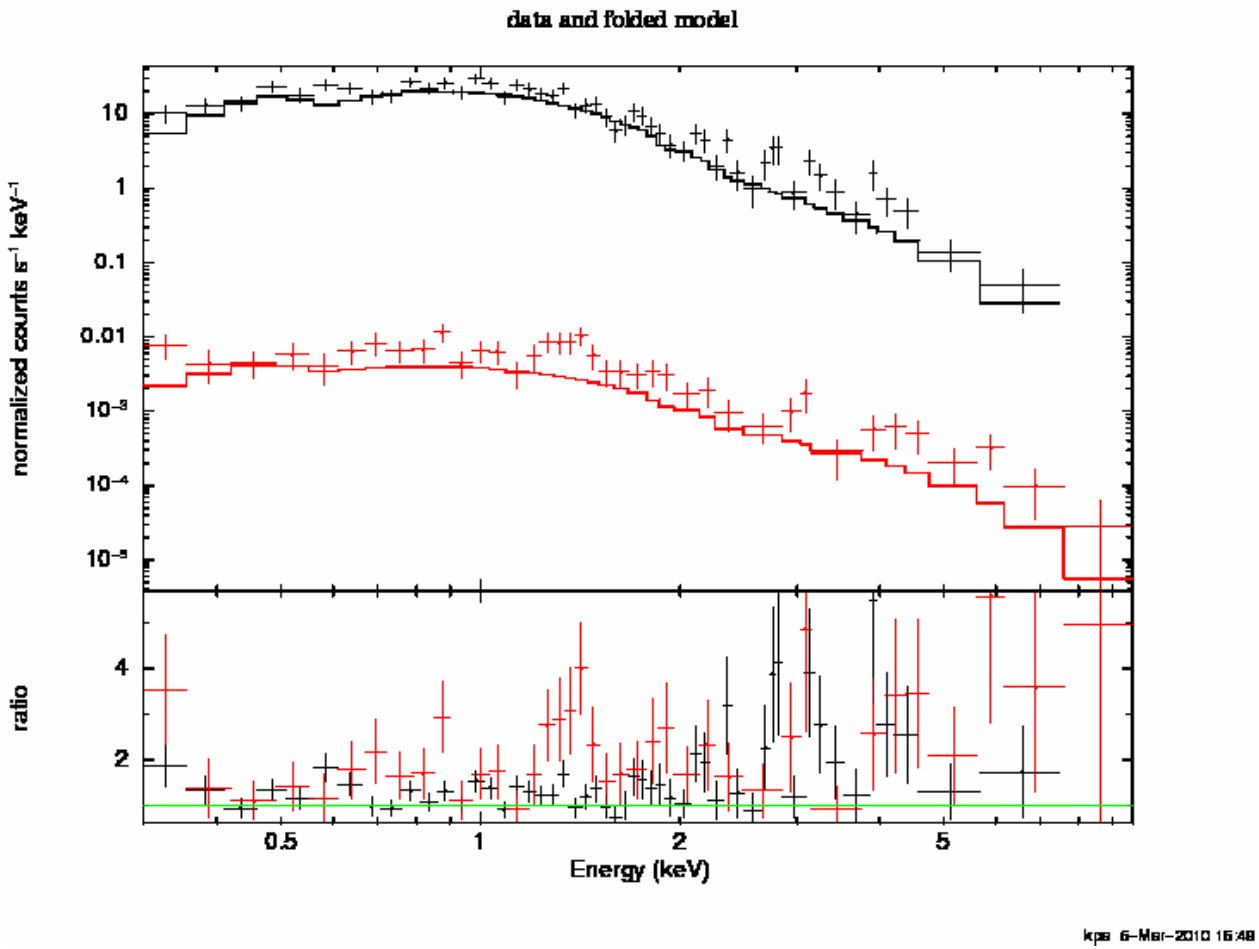
```
========================================================================
Model constant<1>*powerlaw<2>*wabs<3> Source No.: 1   Active/On
Model Model Component  Parameter  Unit     Value
 par  comp
                    Data group: 1
   1    1   constant   factor            1.00000      frozen
   2    2   powerlaw   PhoIndex         1.00000      +/-  0.0
   3    2   powerlaw   norm             1.00000E-04  +/-  0.0
   4    3   wabs       nH       10^22   1.00000      +/-  0.0
                    Data group: 2
   5    1   constant   factor            1.00000      +/-  0.0
   6    2   powerlaw   PhoIndex         1.00000      = 1:powerlaw[2]:PhoIndex
   7    2   powerlaw   norm             1.00000E-04  = 1:powerlaw[2]:norm
   8    3   wabs       nH       10^22   1.00000      = 1:wabs[3]:nH
_____________________________________________________________________________

 Chi-Squared =        2311.68 using 450 PHA bins.
 Reduced chi-squared =        5.18313 for    446 degrees of freedom
 Null hypothesis probability =  1.075949e-248
 Current data and model not fit yet.
```

XSPEC12>stati cstat i.e. use Cash statistics; default is χ^2

XSPEC12>fit

XSPEC - 5


data and folded model
normalized counts s⁻¹ keV⁻¹
10
1
0.1
0.01
10⁻³
10⁻⁴
10⁻⁵
ratio
4
2
0.5
1
2
5
Energy (keV)
kps 6-Mar-2010 16:48

Summary

This was only a very quick tour of the basics of some of the data analysis we can perform for GRBs – we really did only scratch the surface! However, hopefully it will serve as a basic introduction to what is possible…

A short exercise sheet (complete with model answers!) is available, if you would like to have a go at some Swift analysis:

http://www.star.le.ac.uk/~kpa/worksheet.pdf

Useful links

http://www.swift.ac.uk/BAT.shtml - BAT analysis threads

http://www.swift.ac.uk/XRT.shtml - XRT analysis threads

http://www.swift.ac.uk/UVOT.shtml - UVOT analysis threads

http://www.swift.ac.uk/access/ql.php and
http://www.swift.ac.uk/swift_portal/archive.php - Swift data access

http://xmm.esac.esa.int/external/xmm_data_analysis - XMM-Newton data analysis

http://xmm.esac.esa.int/sas/current/documentation/threads - XMM threads

http://xmm.esac.esa.int/xsa/index.shtml - XMM data access

http://heasarc.gsfc.nasa.gov/ftools/xselect/node1.html - XSELECT guide

http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec/index.html - XSPEC homepage

http://heasarc.gsfc.nasa.gov/docs/software/ftools/others/qdp/qdp.html - QDP guide

swift-help@star.le.ac.uk