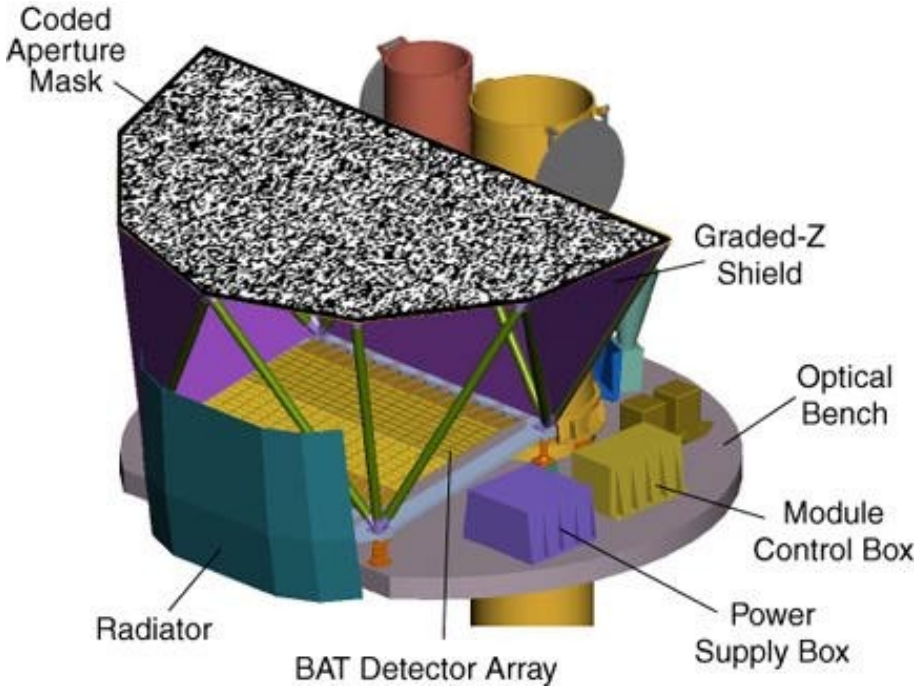
Swift Software Training - BAT

Kim Page

Introduction

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 (also called mask-tagging or ray-tracing) is a way of background-subtracting coded-mask data. Although the files from the SDC 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]bevsh<type>_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!
* or detmask=auxil/sw[obsid]b_qmap.fits and the event file being event/sw[obsid]b_all.evt after running batgrbproduct

This does not produce a new file, but rather adds (or alters) a column called MASK_WEIGHT in the event file.
If the detector mask (map of excluded detectors, since they aren't always all on) is missing, one can be made using bathotpix; see notes at end of presentation.

Images - 1

First, make a detector plane image (DPI). Commands in [ ] are the defaults, but will be remembered if changed to something else:

>batbinevt [weighted=no outunits=counts] detmask=[obsid]/bat/hk/sw[obsid]bdqcb.hk
Input event file name: [obsid]/bat/event/sw[obsid]bevsh<type>_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

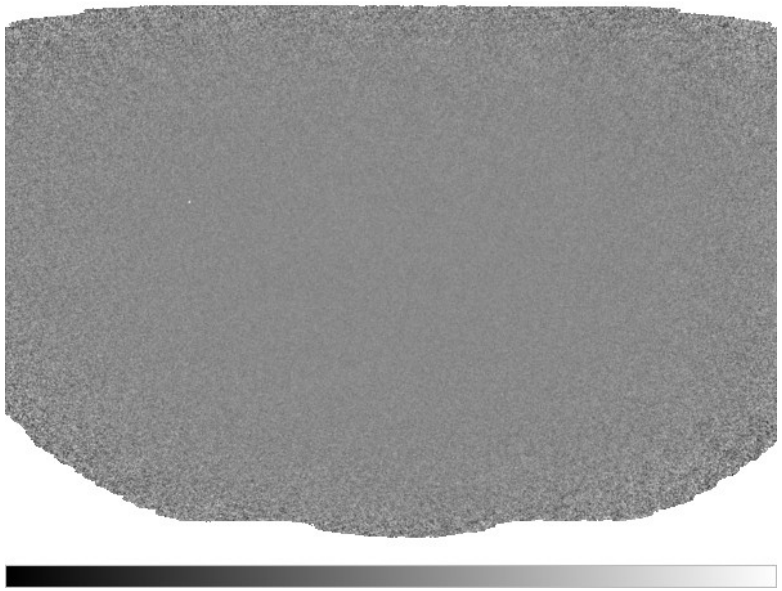
Alternatively:
Energy bin list: 15-25, 25-50, 50-100, 100-350

* or detmask=auxil/sw[obsid]b_qmap.fits and the event file being event/sw[obsid]b_all.evt after running batgrbproduct

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

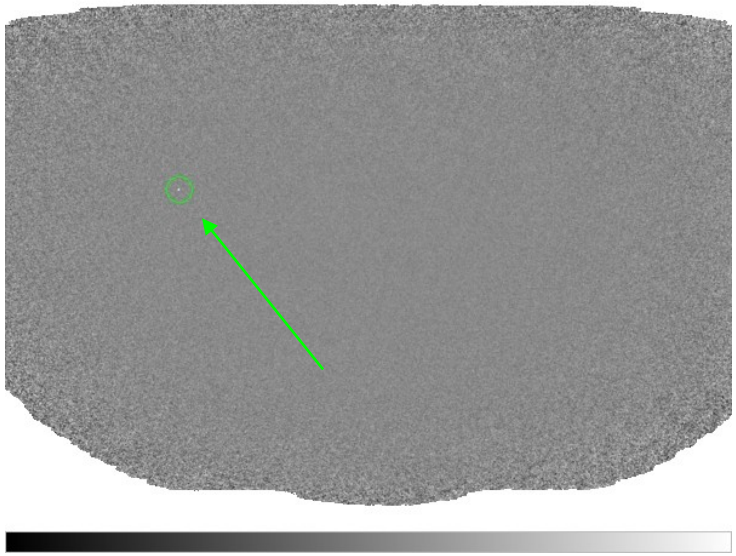
To detect sources, run the simple task batcelldetect.

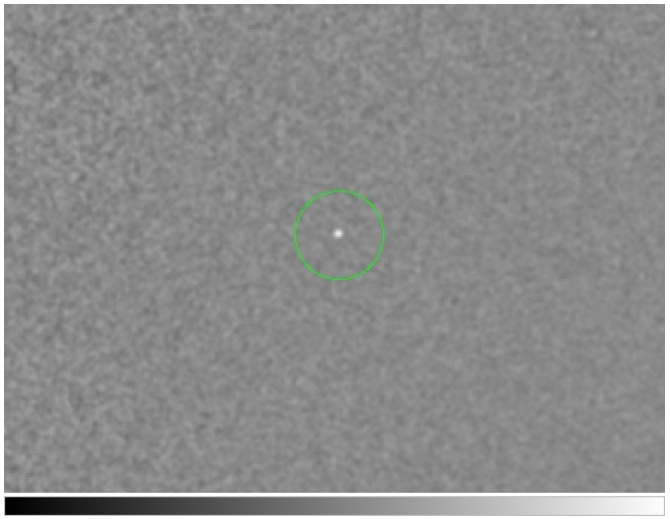
>batcelldetect

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]/hk/sw[obsid]bdqcb.hk *
Input event file name: [obsid]/bat/event/sw[obsid]bevsh<type>_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.

* or detmask=auxil/sw[obsid]b_qmap.fits and the event file being event/sw[obsid]b_all.evt after running batgrbproduct

Light-curves - 2

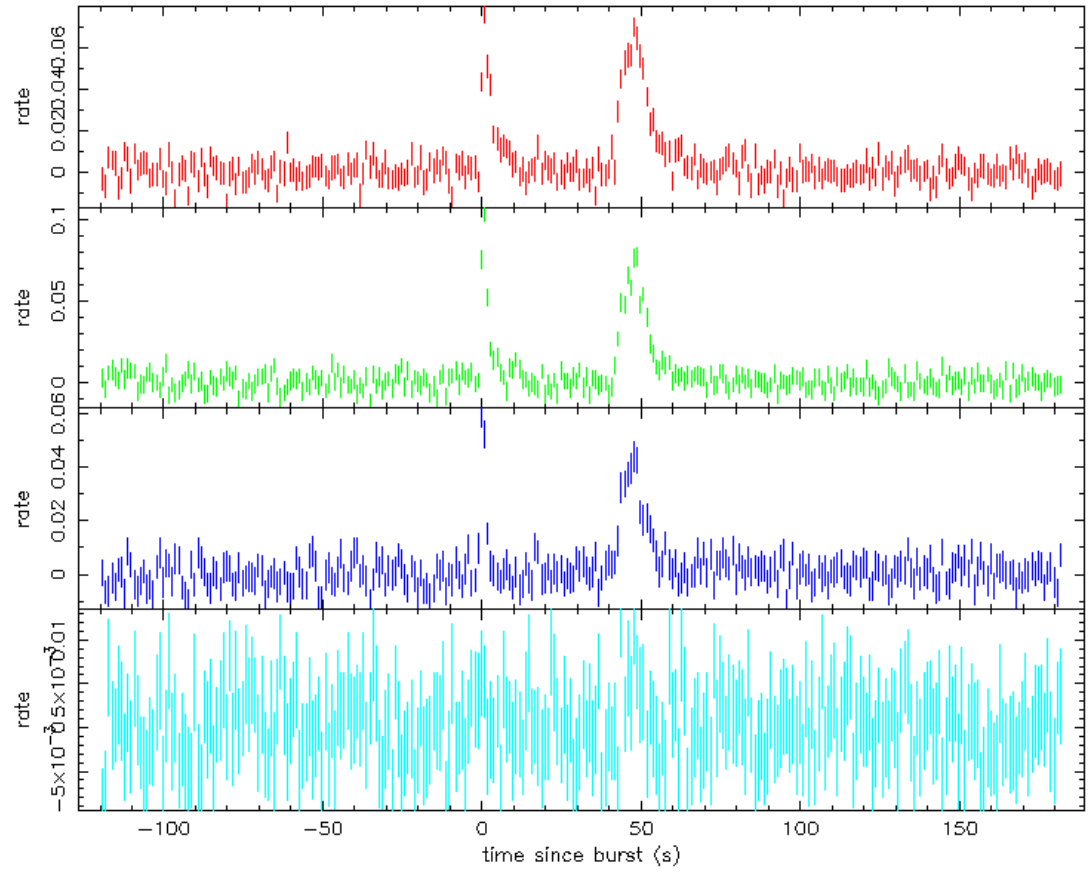
If you have a multi-vector light-curve (i.e., more than 1 energy band), they can be plotted as follows:

>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

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.

and you get…

Light-curves - 3



15-25 keV

25-50 keV

50-100 keV

100-350 keV

Timing - 1

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 T90, T50, TXX, 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.

Timing - 2

Slightly more advanced timing filters can be made using maketime. An example is shown here:

> maketime

Name of FITS file and [ext#]:sw[obsid]sat.fits[ACS_DATA]
Name of output FITS:slew.gti
Selection Expression: FLAGS==b00xxxxxxx or bx1xxxxxxx for settling
Flag, yes if HK format is compact: no
Column containing HK parameter times: time

Since the *sat.fits files tend to contain more than 1 orbit, ftcopy should be used to make sure only the 1st set of slew data are used:

> ftcopy slew.gti[STDGTI]['START>tstart && STOP<tstop']

Output file name:slew_new.gti

Spectra - 1

>batbinevt detmask=[obsid]/hk/sw[obsid]bdqcb.hk *
Input event file name: [obsid]/bat/event/sw[obsid]bevsh<type>_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 defined in the Calibration Database. Note that the energy bins can be supplied in a FITS file. An example of how to make a FITS file is given here:
http://swift.gsfc.nasa.gov/docs/swift/analysis/threads/batlightcurvethread.html

If you want a spectrum for a specific time interval, include tstart/tstop on the command line, in terms of MET.

* or detmask=auxil/sw[obsid]b_qmap.fits and the event file being event/sw[obsid]b_all.evt after running batgrbproduct

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 *
* Raytracing file is auxil/sw[obsid]b_all.evaux after running batgrbproduct

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 build, 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 used 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]bevsh<type>_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.

Things to check - 2

>bateconvert
Input file name: [obsid]/event/sw[obsid]bevsh<type>_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

* The event file is event/sw[obsid]b_all.evt after running batgrbproduct.

If the gain/offset file is missing, the next earliest should be obtained from the trend data on the HEASARC ftp site (ftp://heasarc.gsfc.nasa.gov). Trend data are sorted by months, so look in /swift/data/trend/YYYY_MM/bat/bgainoffs. The files are called sw[obsid]bcbo*.fits.

Things to check - 3

If no detector mask (*bdqcb.hk or similar name – they seem to change them occasionally!) 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 now 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! ☺