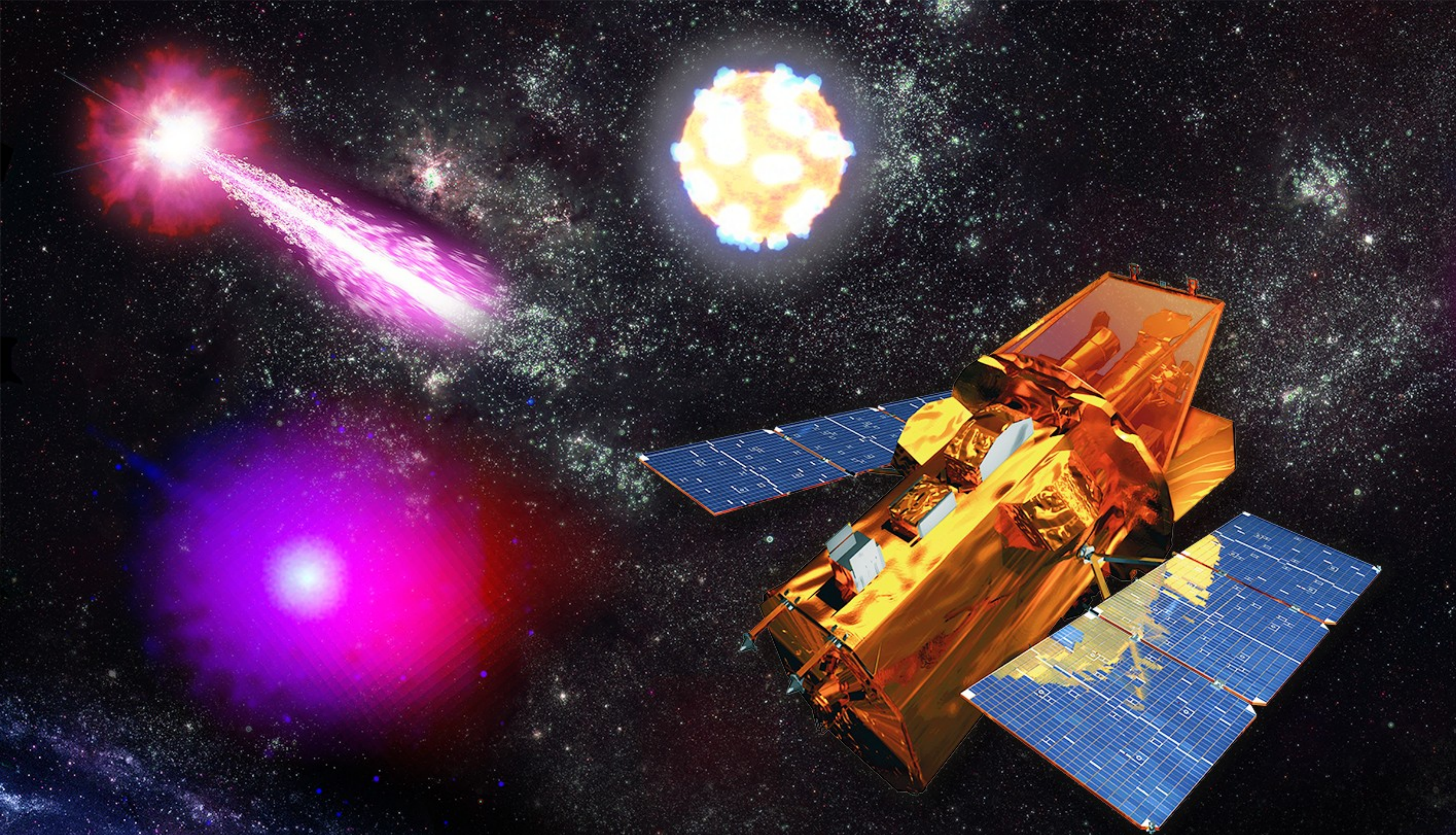


Swift Data Analysis

Kim Page & Andy Beardmore
University of Leicester

Credit: NASA/GSFC

Swift - 1

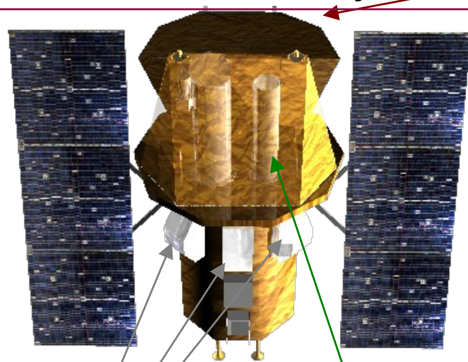
Swift was launched on 20th November 2004, with the main goal of detecting and monitoring Gamma-Ray Bursts.

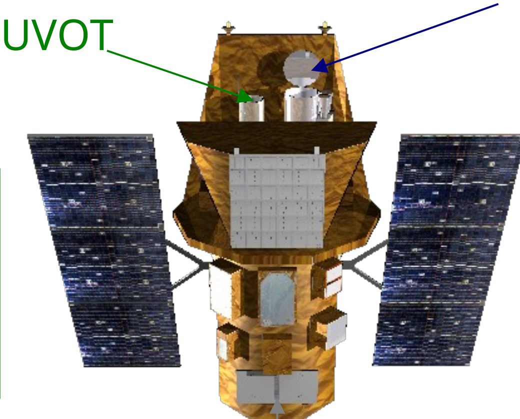
The nominal lifetime of Swift at launch was 2 years, hopefully 5 – and we're now almost at 21 years!

Swift slews around 100 times every day, and the spacecraft has orbited the Earth ~115,000 times so far!

Swift - 2

BAT : 15-350 keV; 2sr FOV; few arcmin accuracy





XRT: 0.3-10keV; 20 arcmin FOV; 3.5 arcsec accuracy

UVOT: UBV/UV filters + optical grisms; sub-arcsec accuracy

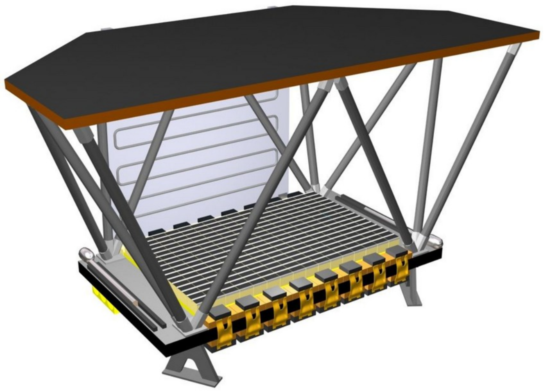
Swift operates autonomously – decides when/how to slew when BAT triggers

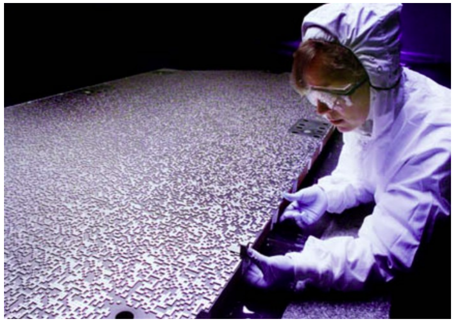
BAT + XRT/UVOT prompt data telemetred rapidly

Full data available within hours

BAT

- Hard X-ray imaging (15-150 keV)
  - Coded mask with 1 mm lead tiles, 1 meter focal length
  - 32,768 CZT detectors launched





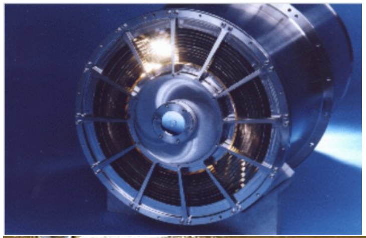
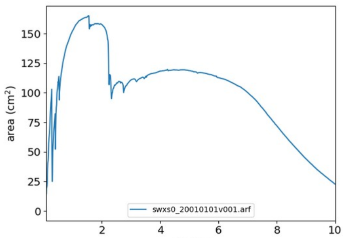
Status

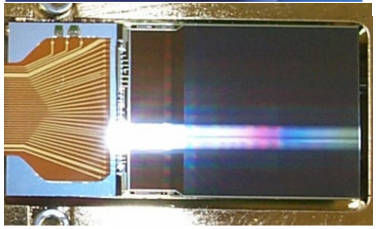
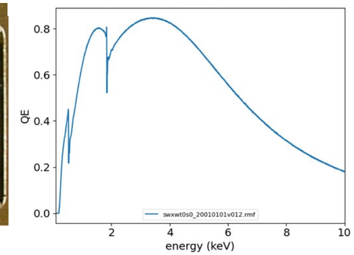
- BAT continues to operate nominally and detect GRBs
- Typically 12,000 - 18,000 detectors enabled, others disabled due to noise
- 3 of 4 redundant loop heat pipe controllers have failed, but 1 remaining controller is sufficient to maintain thermal requirements

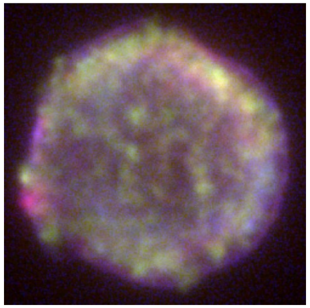
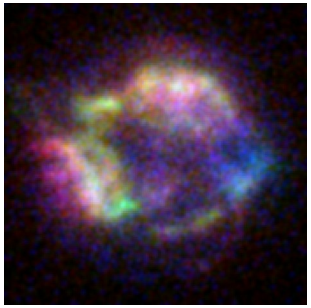
XRT

- Comprises
  - Wolter Type I mirror (Jet-X flight spare)
  - e2v CCD22 detector (same as XMM MOS)
  - Energy range 0.3-10 keV

- XRT continues to operate well
  - No new anomalies to report
  - Last reboot back in 2016
  - Despite failure of TEC in 2004, CCD thermal control is still well maintained in the range -65C to -50C by careful science target planning

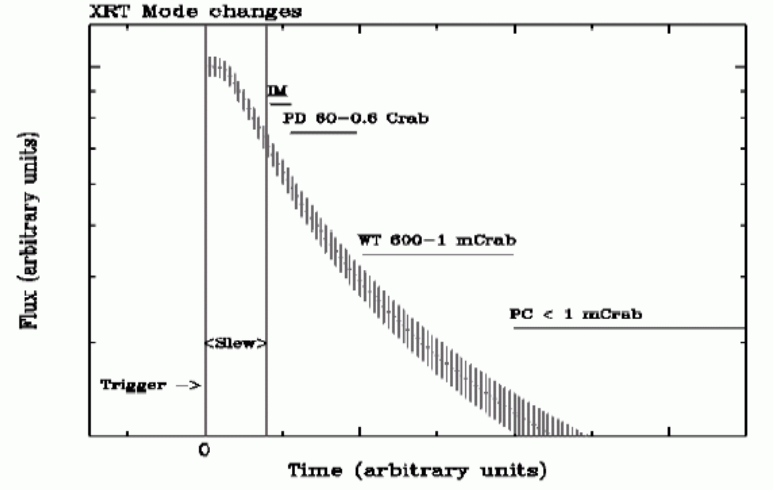
XRT Modes

Imaging mode

- Only used immediately after BAT trigger
- Exposure of 0.1 or 2.5 s

Photo-Diode mode (Low-Rate and Piled-Up) -

- Disabled following 2005 micrometeoroid strike
- No spatial information
- 0.14 ms time resolution



Windowed Timing mode (CR ≥ 2 count s^{-1})

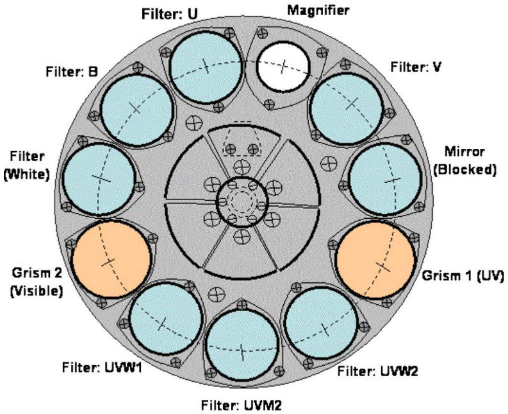
- 1-dimensional imaging
- 1.7 ms time resolution

Photon Counting mode

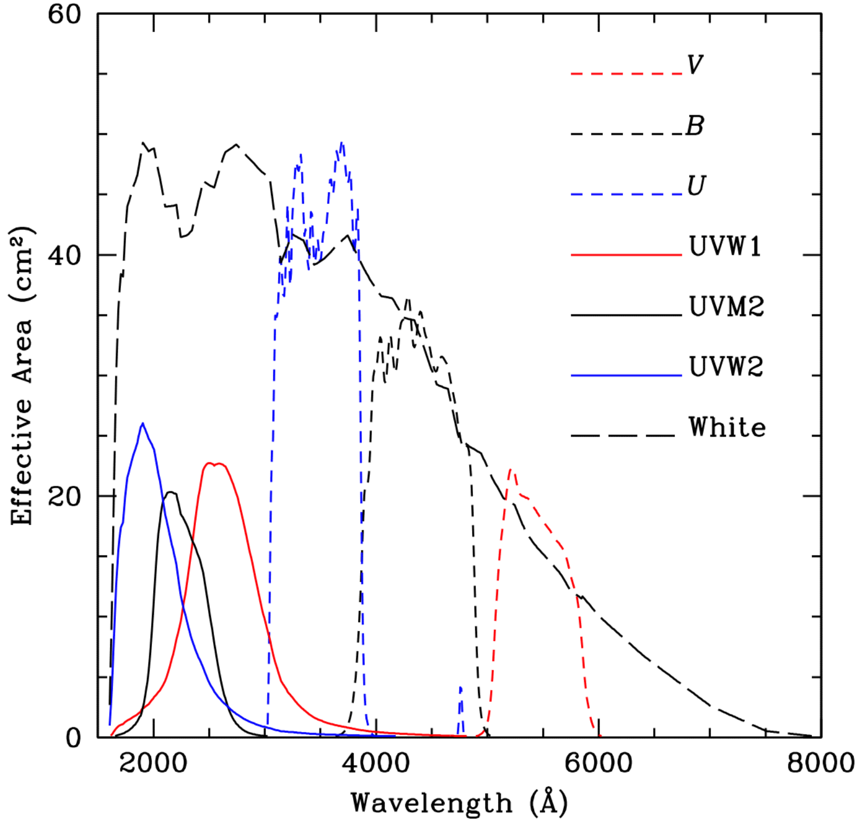
- 2.5 s time resolution

UVOT

- 30 cm modified Ritchey-Chrétien optical telescope
- Imaging and grism spectroscopy over 170-600 nm
- Microchannel Plate intensified CCD detector
- UBV and UV filters



UVOT Filters

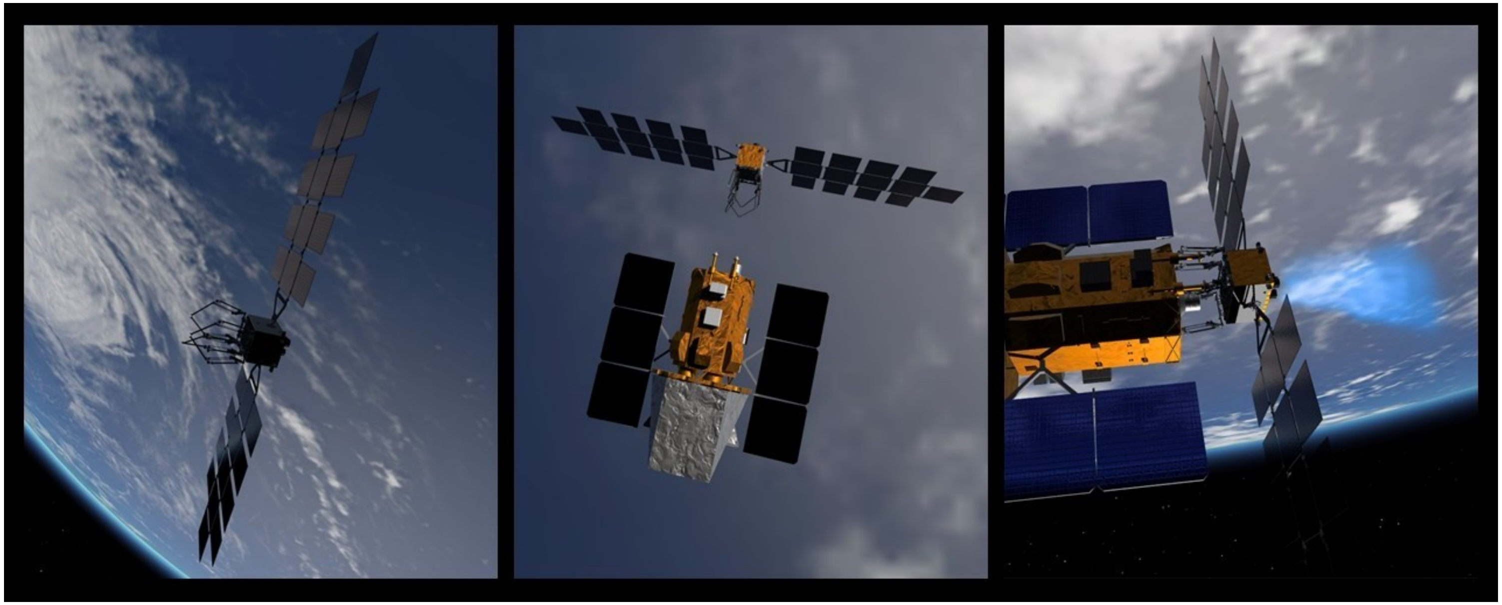


Filter	Central Wavelength (Å)
v	5468
b	4392
u	3465
uvw1	2600
uvm2	2246
uvw2	1928

Note that uvw1 and uvw2 have extended red (longer wavelength) tails, so will be somewhat contaminated by the source optical emission; uvm2 does not suffer from this problem.

In the News - Swift Orbit Boost

- Due to recent Solar activity, Swift's orbital decay rate has increased
  - Predicted re-entry towards end of 2026
- NASA have recently given the go ahead for Katalyst Space Technologies to build a probe to catch Swift and give it an orbital boost.



Useful Titbits

Swift is in a low-Earth orbit (~90 minutes; ~15 orbits per day).

Observations are made up of one or more continuous on-target pointings known as "snapshots".

For spacecraft safety, snapshots are scheduled to be **no longer** than 1.8 ks.

There are online visibility checkers, to determine when your source of interest is observable (not too close to the Sun, Moon, orbital pole etc):

https://www.swift.psu.edu/operations/roll.php (day by day visibility)

https://www.swift.psu.edu/operations/visibility.php (snapshot visibility)

Beware optical loading! Bright optical sources can cause significant charge in the CCD, leading to spurious "X-ray" events. If the source is brighter than about 10^{th} mag (8^{th} for very blue optical sources), use WT mode instead (which should be OK up to ~3^{rd} mag).

https://www.swift.ac.uk/analysis/xrt/optical_loading.php

Data Access

Data collected by Swift are promptly (typically within a few hours of the data being collected) available on the Quick Look site, and the observations remain there for about a week, before moving to the archive.

Since Swift observations frequently consist of multiple snapshots, ObsIDs can be updated as more data are collected (indicated by the version number).
https://www.swift.ac.uk/archive/ql.php?sortcol=observed
https://swift.ssdc.asi.it/cgi-bin/ql/createqlhtml?sortcol=Observed
https://swift.gsfc.nasa.gov/sdc/ql/

After a week or so (at which point the ObsIDs should definitely be completed), the data move into the archive:
https://www.swift.ac.uk/swift_live/index.php
https://www.ssdc.asi.it/mmia/index.php?mission=swiftmastr
https://heasarc.gsfc.nasa.gov/cgi-bin/W3Browse/swift.pl

The top ObsID directory will contain sub-directories for each instrument and auxiliary data.

XRT Product Generator - 1

The University of Leicester provides an online XRT product generator (for light-curves, spectra, images, positions and source detection), which can either be accessed via a web form at
https://www.swift.ac.uk/user_objects/
or through the swifttools API
https://www.swift.ac.uk/API/

GRBs (both BAT triggers and those we follow up from other missions) are automatically analysed, with the results promptly available (and updated whenever we collect more data) at
https://www.swift.ac.uk/xrt_products/

Similarly, if we follow up LVK GW triggers, the results are provided online (analysed automatically, but the sources are vetted by humans):
https://www.swift.ac.uk/LVC

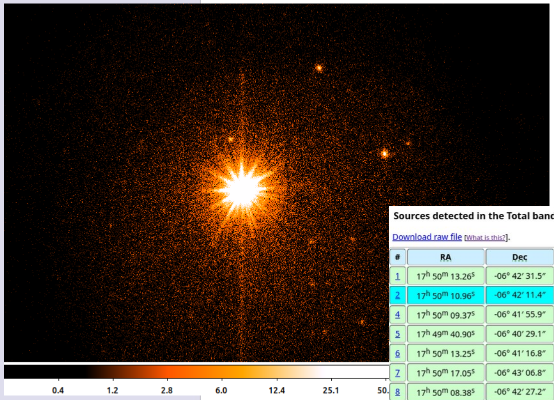
We also have the Living Swift XRT Point Source Catalogue (LSXPS):
https://www.swift.ac.uk/LSXPS

XRT Product Generator - 2

There is online documentation, with even more details in
Evans et al. (2007, A&A, 469, 379) - light-curves
Evans et al. (2009, MNRAS, 397, 1177) – spectra and enhanced positions
Evans et al. (2023, MNRAS, 518, 174) - transient detection

These products are completely suitable for scientific analysis (of point sources) and the tools are relied upon by the Swift team on a daily basis.

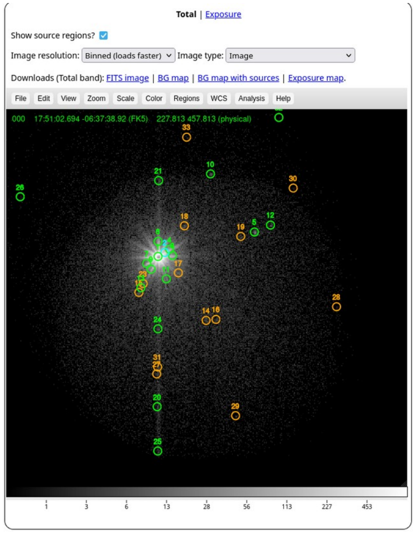
XRT Product Generator - 3

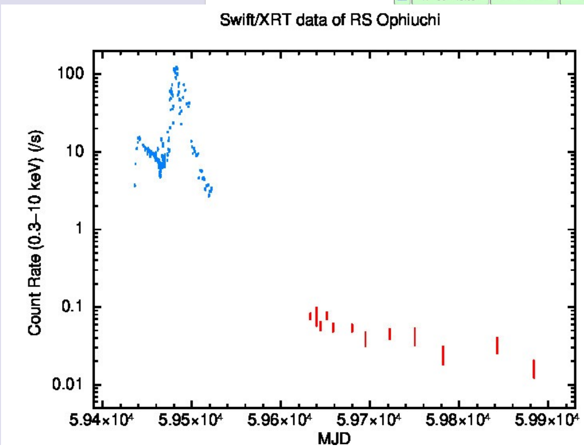



Sources detected in the Total band

Download raw file [What is this?]

#	RA	Dec	Det flag
1	17h 50m 13.26s	-06° 42′ 31.5″	Good
2	17h 50m 10.96s	-06° 42′ 11.4″	Reasonable
4	17h 50m 09.37s	-06° 41′ 55.9″	Good
5	17h 49m 40.90s	-06° 40′ 29.1″	Good
6	17h 50m 13.25s	-06° 41′ 16.8″	Good
7	17h 50m 17.05s	-06° 43′ 06.8″	Good
8	17h 50m 08.38s	-06° 42′ 27.2″	Good
9	17h 50m 15.57s	-06° 43′ 35.0″	Good
10	17h 49m 55.65s	-06° 35′ 38.8″	Good
11	17h 50m 10.47s	-06° 44′ 24.0″	Good
12	17h 49m 35.53s	-06° 39′ 54.6″	Good
13	17h 50m 19.01s	-06° 45′ 08.0″	Good
14	17h 49m 57.12s	-06° 47′ 51.6″	Poor
15	17h 50m 19.70s	-06° 45′ 31.6″	Poor
16	17h 49m 53.81s	-06° 47′ 46.1″	Poor
17	17h 50m 06.42s	-06° 43′ 53.8″	Poor
18	17h 50m 04.38s	-06° 39′ 58.5″	Poor
19	17h 49m 45.48s	-06° 40′ 51.7″	Poor
20	17h 50m 13.52s	-06° 55′ 02.5″	Good
21	17h 50m 13.05s	-06° 36′ 12.5″	Good





Running the XRT pipeline - 1

If you wish to extract data products “by hand”, then the command required is:

xrtpipeline cleanup=no >& log

Notes:

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.

Including the datamode input (e.g. datamode=pc) on the xrtpipeline command line will just process the mode stated (pc – Photon Counting); wt – Windowed Timing)

clobber=no is the default.

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. Default is usually fine (and safest!), though.

All other required inputs will be prompted for, if not included on the command line.

Running the XRT pipeline - 2

Using GRB 250504A as an example (dataset 01310284000); note that the output directory cannot be a sub-directory of the input one.

```
==========================================================
               Running SWIFT XRT pipeline
 Task: xrtpipeline Version: 0.13.7 Release Date: 2022-02-14
==========================================================
Source RA position (degrees or hh mm ss.s) or POINT or OBJECT [ ] : 269.61634

Source DEC position (degrees or dd mm ss.s) or POINT or OBJECT [ ] : -40.36676

Target Archive Directory Path [ ] : 01310284000

Stem for FITS input files [i.e. sw00000000000] [ ] : sw01310284000

Directory for outputs [ ] : 01310284000-xrt
```

Pipeline Files

A large number of files will be generated. We do NOT recommend using these spectra/light-curves, since they are extracted using default regions, no background files are created, and there is no accounting for pile-up etc. It is possible to skip this stage by including extractproducts=no on the command line.

Example useful files:

x for XRT

Event lists: sw01310284000xwtw2po_cl.evt and sw01310284000xpcw3po_cl.evt

11 digit ObsID, consisting of 8 digit target ID/trigger number and 3 digit segment number.
The target ID is unique to a source (though any given source may have multiple target IDs!)

wt or pc

w2, w3 or w4 gives the window size – slight change in number of pixels.

po = pointing (ie on target); ignore sl and st (slew and settling).

cl = cleaned (level 2) event list. Also present: uf (and ufre for WT) = unfiltered (level 1, or raw) data.

Region files: sw01310284000xwtw2po.reg and sw01310284000xpcw3po.reg

(assuming xrtpipeline was run with cleanup=no)

XSELECT - 1

To extract images, spectra or light-curves, XSELECT should be used.

> xselect

> Enter session name >[xsel]

> read event

> Enter the Event file dir >[ ] ./

> Enter Event file list >[ ] sw01310284000xwtw2po_cl.evt

Got new mission: SWIFT

> Reset the mission ? >[yes]

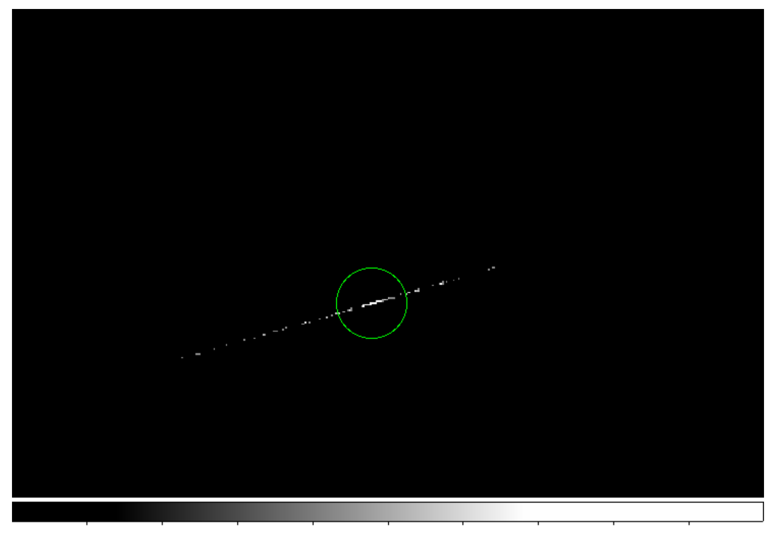
```
  OBJECT                   OBS_ID      DATE-OBS                 DATAMODE
1 Burst (269.617, -40.366) 01310284000 2025-05-04T23:28:58.7007 WINDOWED
```

XSELECT - 2

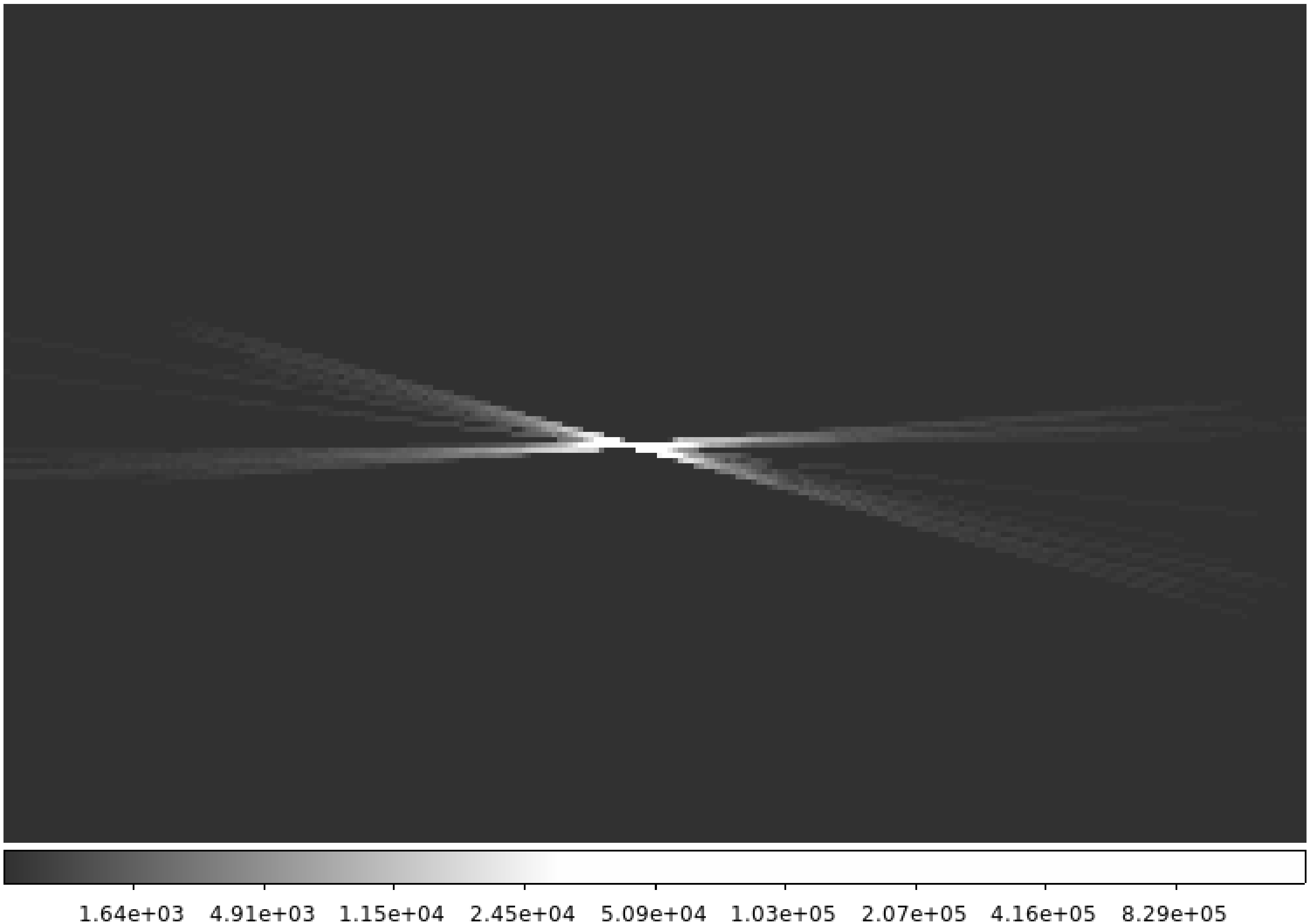
> extract image

```
         Total       Good    Bad: Time      Phase      Grade       Cut
           334        334           0          0          0          0
===============================================================================
    Grand Total       Good    Bad: Time      Phase      Grade       Cut
           334        334           0          0          0          0
   in  53.064     seconds
 Image              has       334 counts for    6.294      counts/sec
Keyword TLM2FITS has two values: 'V6.4' and 'V6.4     '
xsel3059262:SWIFT-XRT-WINDOWED >
```

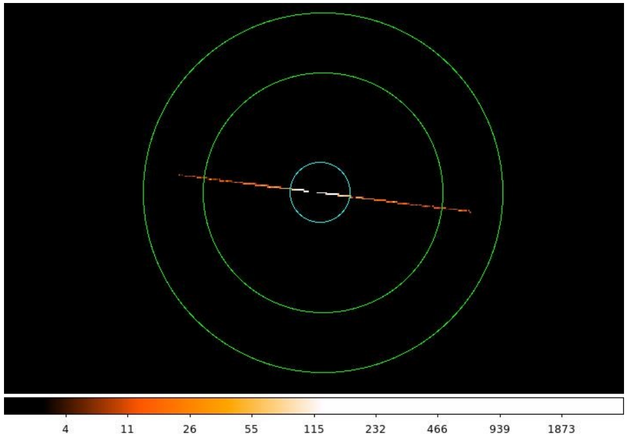
> plot image – opens in ds9. Source and background regions can be defined and saved.

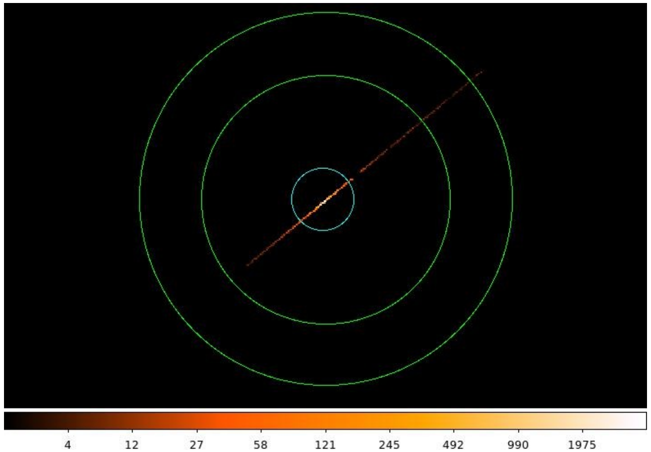


Roll Angle Spread



WT Background





If you have multiple WT observations of the same source, at different roll angles, this is a useful method for setting up a single background region which can always be used.

Define a large annulus, where the inner (r_1) and outer (r_2) radii are symmetrical around 100 pixels (e.g., r_1 = 80 and r_2 = 120, or r_1 = 70, r_2 =130). This ensures that, no matter where the source is in the window, the background region will contain $r_2 - r_1 - 1$ pixels (minus 1 since the end-of-window pixels are automatically flagged as bad by the ground processing). Then change the BACKSCAL keyword in the background spectrum to $r_2 - r_1 - 1$ and the source spectrum to $2r_s$ (assuming the source region is a circle with radius r_s); e.g.,

fparkey 39 WTback.pi+1 BACKSCAL

https://www.swift.ac.uk/analysis/xrt/backscal.php

XSELECT - 3

Useful commands within XSELECT:

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

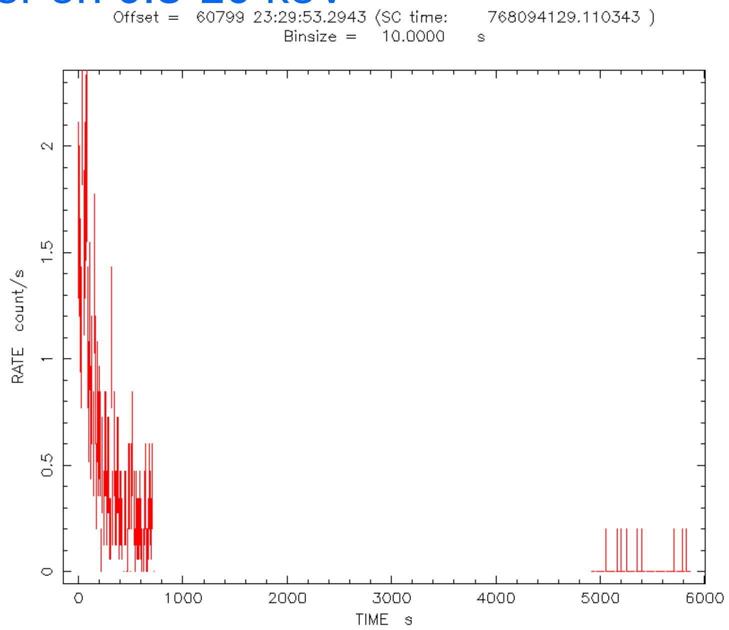
> set binsize 10 – if you want bins of 10 s; type this BEFORE lc extraction! Smallest bin size permitted for WT mode is 0.0018 s, and for PC mode is 2.5074 s. Bins longer than 10 s cannot be directly corrected by xrtlccorr.

> filter pha_cut 30 1000 – if you want to filter on 0.3-10 keV

> 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 - 4

> filter time cursor – 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. seconds since start of observation).

> extract spectrum – uses all previous filtering (grades, regions, times).

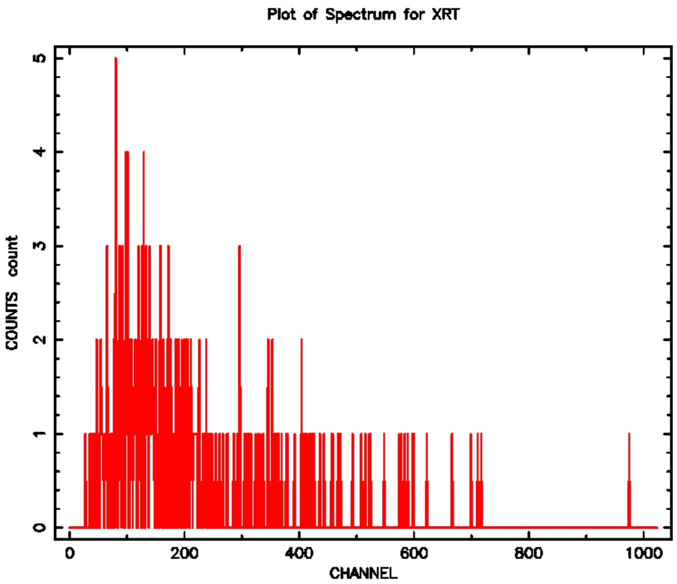
Do NOT include a filter pha_cut command when extracting a spectrum! You won't be able to fit it.

> save spectrum – by default this will be a PI file, suitable for XSPEC and for use with
the RMFs (providing no filtering with pha_cut was performed).

By default, PC data are grades 0-12 and WT are 0-2. Both modes are also calibrated for grade 0 alone (filter grade 0 in XSELECT), as well as 0-4 for PC.

Outside XSELECT, the spectrum should be run through grppha (see later).

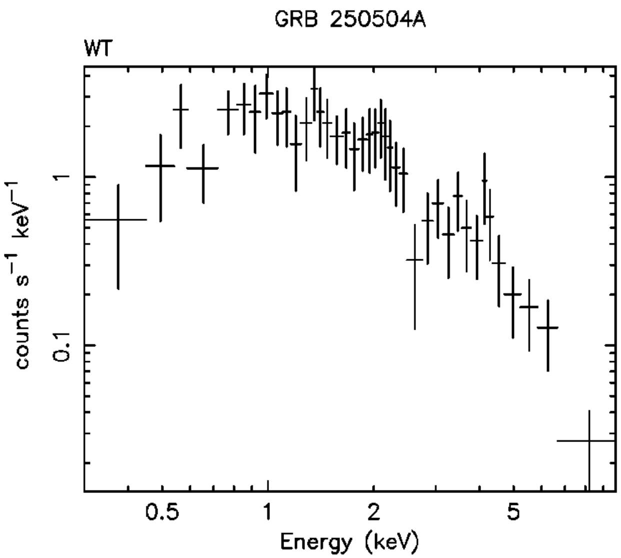
Spectra



Spectrum within XSELECT

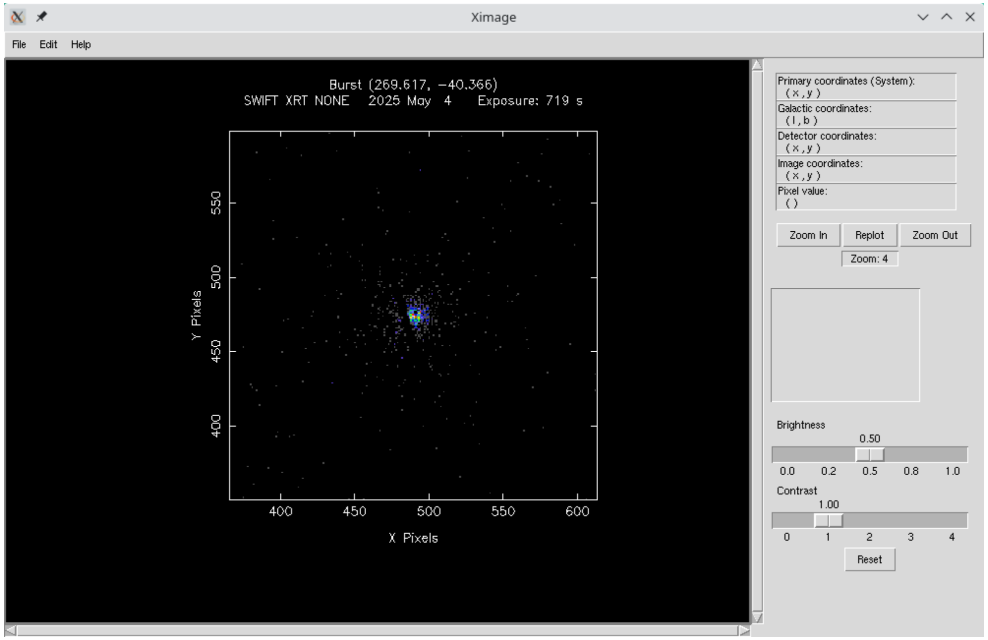
1 channel = 10 eV

Binned with grppha and "prettified" within XSPEC.



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 for PC data is to run the PSF command in XIMAGE:



> ximage

> read PC1.img

> cpd /xtk

> display

> back

> psf/cur

– click in the centre and as far out as you want to determine the PSF (probably ~30-40 pixels, unless **really** bright)

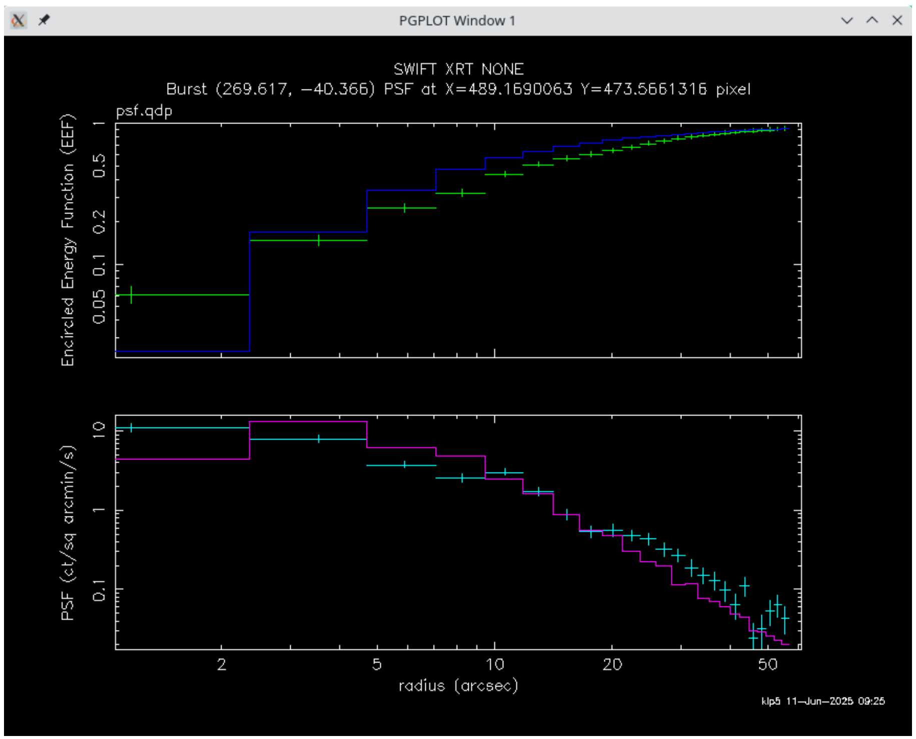
Pile-up - 2

This then takes you into QDP, with the PSF profile plotted; this can be fitted with a King+Gaussian function:

$$\mathbf{PSF(r) = W\ exp[-r^2/(2\sigma^2)] + (1\text{-}W)[\ 1 + (r/r_c)^2\ ]^{-\beta}}$$

where, for PC mode,

$$\mathbf{W \sim 0.075;\ \sigma \sim 7.42;\ r_c \sim 3.72;\ \beta \sim 1.31}$$



Either use the command

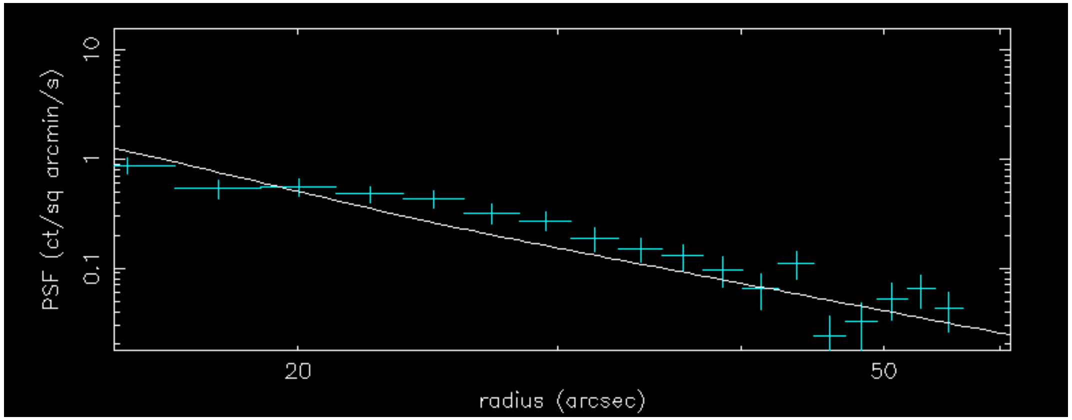
col off 1 2 3 4 6

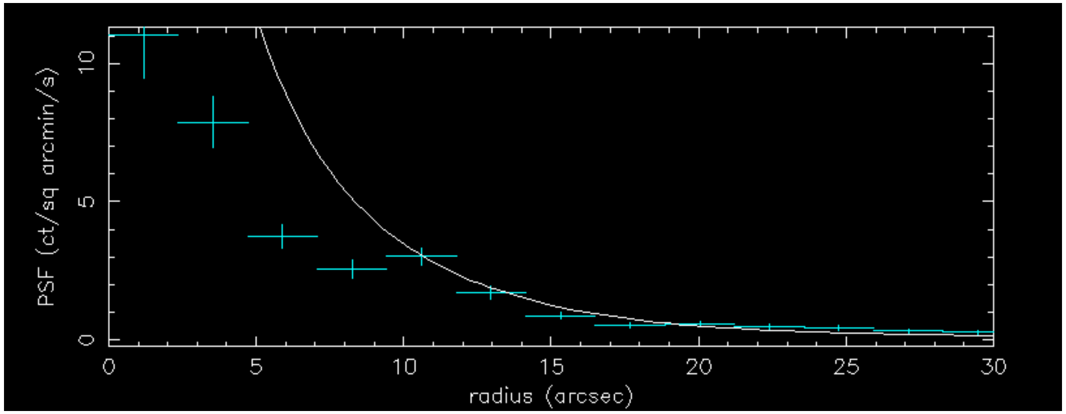
or

fit on 5

The cyan coloured data (dataset 5) points in the lower panel are the only ones we're interested in.

Pile-up - 3





Download the cod file https://www.swift.ac.uk/analysis/xrt/psf.cod) and fit to the wings of the PSF (model psf.cod within QDP), where pile-up should be negligible – white line in these figures.

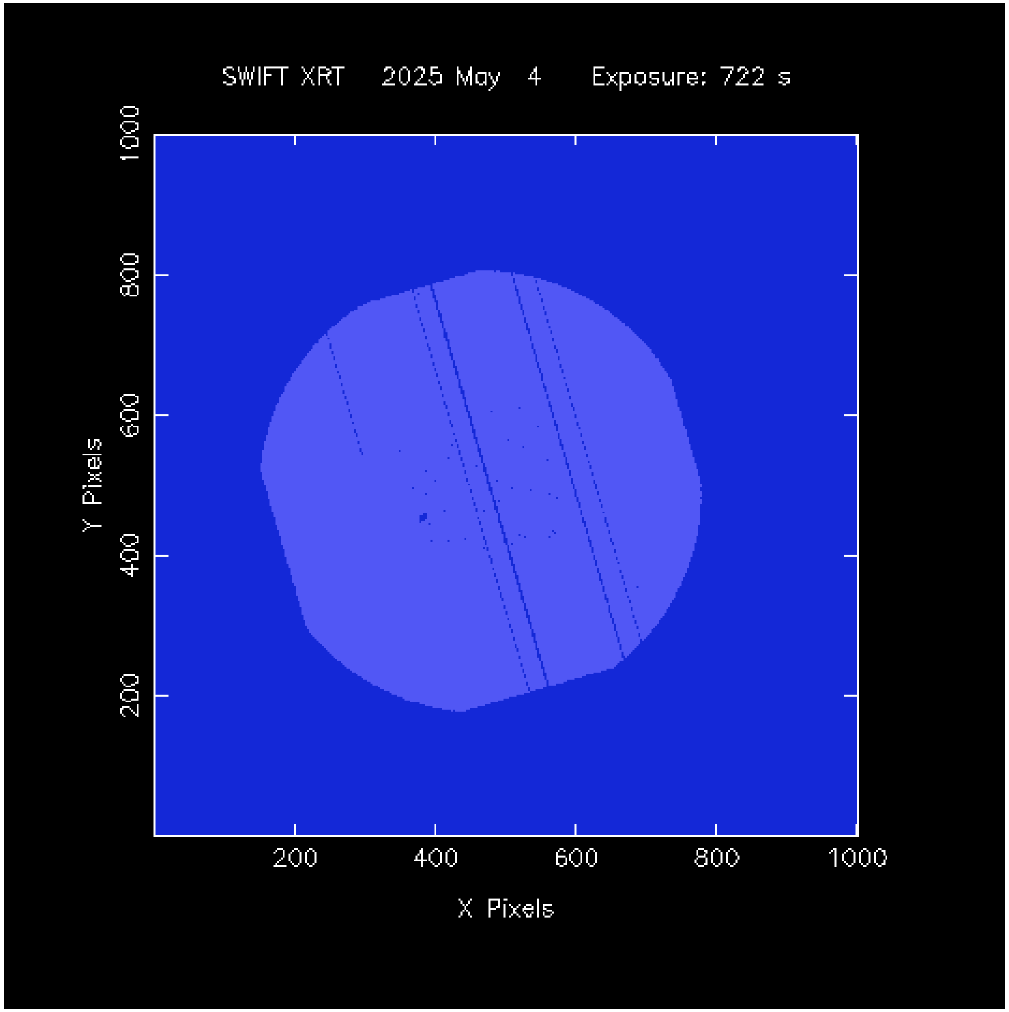
Extrapolating down to smaller radii shows where the model and data start to diverge; this is the radius which should be excluded when extracting spectra/light-curves.

1 arcsec = 2.36 pixels

This is only for PC mode; for WT, just exclude enough of the core until CR < 100-150 count s^{-1}.

See also
http://www.swift.ac.uk/analysis/xrt/pileup.php

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 by xrtpipeline.

If a shorter interval is required:

Use filter time cursor (or scc) 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, assuming you’re working in the 01310284000-xrt directory, this should be run as:

> xrtexpomap attfile=../01310284000/auxil/sw01310284000pat.fits hdfile=../01310284000/xrt/hk/sw01310284000xhd.hk infile=orbit1.evt stemout=orbit1 outdir=./

This will make an exposure map called orbit1_ex.img.

Exposure Maps - 4

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 pat.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:

```
> ximage
> read sw01310284001xpcw3po_ex.img
> read sw01310284002xpcw3po_ex.img
> sum
> save
> read sw01310284003xpcw3po_ex.img
> sum
> save
> write/fits seg001-003_ex.img
```

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

```
> fparkey F seg001-003_ex.img+0 VIGNAPP add=yes
```

Ancillary Response Files - 1

There are 2 main types of XRT calibration file required for spectral fitting: the Redistribution Matrix Files (RMFs) and the Ancillary Response Files (ARFs). The RMFs are located in the Calibration Database (CALDB), and are only dependent on the mode and grade selection used. An ARF has to be created for each spectrum, using the tool xrtmkarf, using a corresponding exposure map. For example:

> xrtmkarf expofile=sw01310284001xpcw3po_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. Avoids mistyping numbers!

If an extended source is being considered, include extended=yes on the command line.

Ancillary Response Files - 2

When an ARF is created, the information written to the screen will include 2 lines saying: PERC. OF FLUENCE WITHIN THE ROI followed by a percentage, one given for ~ 1 keV, the other an average (will be similar), e.g.

PERC. OF FLUENCE WITHIN THE ROI: 73.747015 % (AT 1.002500 keV)

PERC. OF FLUENCE WITHIN THE ROI: 73.745537 % (ON AVERAGE)

This number can be used to estimate the correction factor to account for the PSF losses caused by pile-up and/or exposure map corrections.

The factor required for the loss of counts is 100/<percentage>

(in this example: 100/73.745537 = 1.356).

Note that the ARF will automatically correct the flux/normalisation from a model fitted in XSPEC, but NOT the count rate XSPEC gives; that would need multiplying by this correction factor.

grppha

> grppha

Please enter PHA filename[ ] WT.pi

Please enter output filename[ ] !WT.pi

```
-------------------------
MANDATORY KEYWORDS/VALUES
-------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
EXTNAME    - SPECTRUM        Name of this BINTABLE
TELESCOP   - SWIFT           Mission/Satellite name
INSTRUME   - XRT             Instrument/Detector
FILTER     - NONE            Instrument filter in use
EXPOSURE   - 909.01          Integration time (in secs) of PHA data
AREASCAL   - 1.0000          Area scaling factor
BACKSCAL   - 1.25200E-03     Background scaling factor
BACKFILE   - none            Associated background file
CORRSCAL   - 1.0000          Correlation scaling factor
CORRFILE   - none            Associated correlation file
RESPFILE   - none            Associated redistribution matrix file
ANCRFILE   - none            Associated ancillary response file
POISSERR   - TRUE            Whether Poissonian errors apply
CHANTYPE   - PI              Whether channels have been corrected
TLMIN1     - 0               First legal Detector channel
DETCHANS   - 1024            No. of legal detector channels
NCHAN      - 1024            No. of detector channels in dataset
PHAVERSN   - 1.2.0           OGIP FITS version number
STAT_ERR   - FALSE           Statistical Error
SYS_ERR    - FALSE           Fractional Systematic Error
QUALITY    - TRUE            Quality Flag
GROUPING   - FALSE           Grouping Flag
--------------------------------------------------------------------
--------------------------------------------------------------------
```

GRPPHA [ ] bad 0-29 && group min 1

GRPPHA [ ] chkey respfile swxwt0to2s6_20210101v017.rmf

GRPPHA [ ] chkey ancrfile WT_exp.arf

GRPPHA [ ] chkey backfile WTback.pi

GRPPHA [ ] exit

If you wish to over-write the same file name, precede with !

Use exit to leave grppha and save your changes; quit will cancel the changes.

We recommend fitting with Cash statistics, for which XSPEC needs “group min 1”.

Light-Curve Corrections

There is a useful task called xrtlccorr which calculates the bad column correction factor for every individual orbit and corrects the corresponding light-curves. This is the lc equivalent of using an ARF for the spectrum.

> 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 : sw0131028400pat.fits

Name of the output Instrument Map File : DEFAULT

Name of the input Event FITS file : sw0131028400xpcw3po_cl.evt

Name of the input Housekeeping Header Packets FITS file : sw0131028400xhd.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.

Light-Curve Subtraction

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.

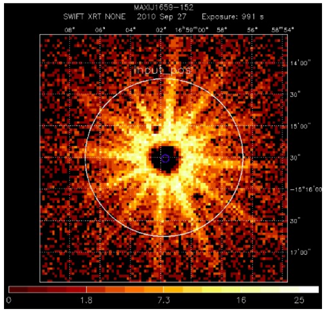
Scaling factor for background [ ] 0.25

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 circular source region has a radius of 30 pixels and the background 60 pixels, the background scaling factor would be $\pi 30^2/\pi 60^2 = 0.25$

X-ray Timing Analysis - 1

Basic timing analysis using XRT data from MAXIJ1659-152 (WT data from ObsID 00434928000):



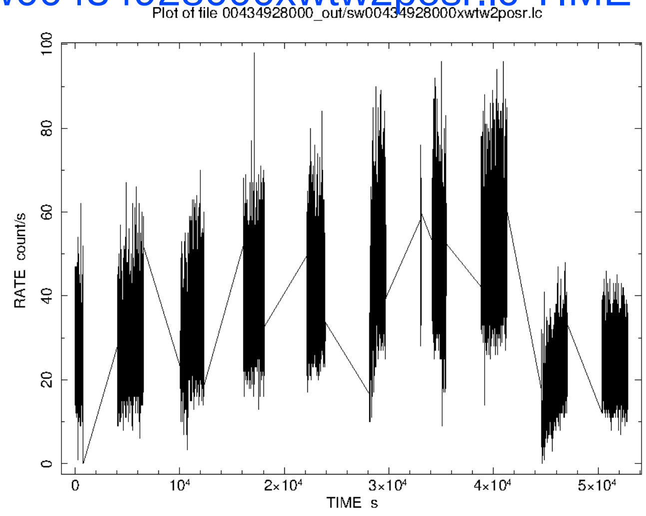
PC mode image from OBSID 00434928004

1. Initial processing using the PC position from user_objects:

xrtpipeline indir=00434928000 outdir=00434928000_out steminputs=sw00434928000 obsmode=POINTING datamode=WT srcra=254.7572 srcdec=-15.2587

2. Examine default light curve to see what we have with using *fplot*:

fplot 00434928000_out/sw00434928000xwtw2posr.lc TIME RATE '-' /xs '' offset=y



X-ray Timing Analysis - 2

3. Adjust event arrival times to solar system barycentre using *barycorr* – essential for short timescale, periodic sources (e.g. pulsars) - requires the satellite attitude and orbit (sao) file and source RA, Dec:

barycorr infile=00434928000_out/sw00434928000xwtw2po_cl.evt
outfile=00434928000_out/sw00434928000xwtw2po_cl_bcorr.evt
orbitfiles=00434928000/auxil/sw00434928000sao.fits.gz ra=254.7572 dec=-15.2587

4. Extract the source light curve (using a binsize 10x native WT readout time, grade 0 events and PI range 30-1000) for the first snapshot. See XSELECT-3 slide, or using *extractor* directly:

extractor filename=00434928000_out/sw00434928000xwtw2po_cl_bcorr.evt[PI=30:1000]
eventsout=NONE regionfile=src_c30.reg qdpfile=NONE
fitsbinlc=00434928000_out/sw00434928000xwtw2po_cl_bcorr_t0.0179s_orb1.lc
unbinlc=NONE phafile=NONE imgfile=NONE fullimage=Y ecol=PI xcolf=X ycolf=Y xcolh=X
ycolh=Y binf=1 binh=1 xfkey=TLMAX yfkey=TLMAX xhkey=TLMAX yhkey=TLMAX gtinam=GTI
gcol=GRADE gstring=0 tcol=TIME timefile=00434928000_out/orb1.gti binlc=0.17790999E-01
lcthresh=0.1 clobber=Y

Here, orb1.gti contains the barycentric start stop times of the first snapshot:

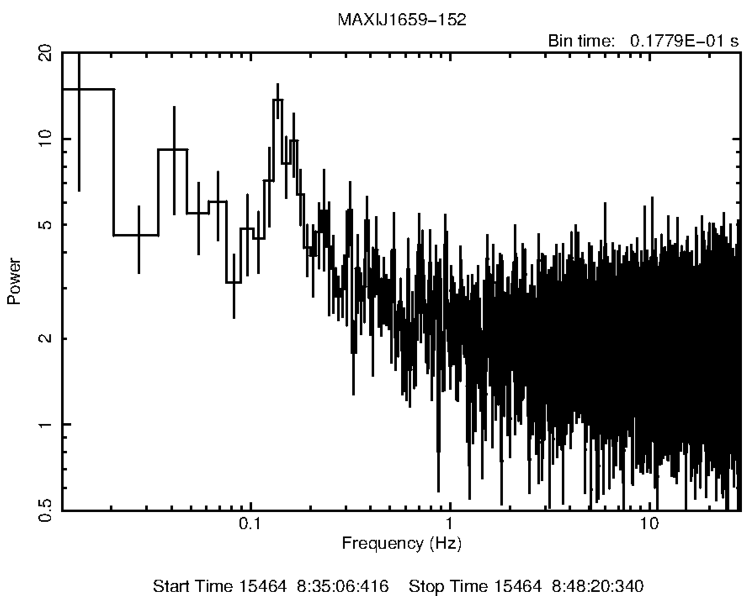
307096442.2   307097236.2

X-ray Timing Analysis - 3

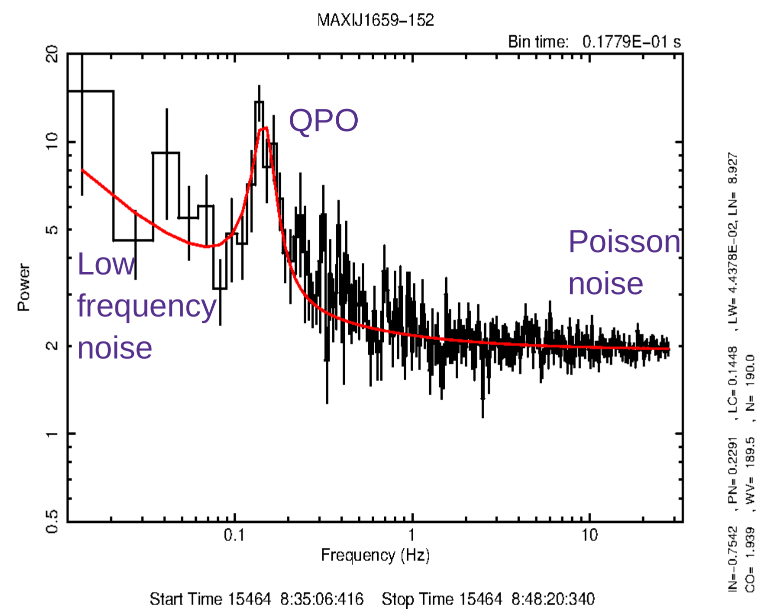
5. Generate average powerspectrum from (11) intervals of 4096 time bins using powspec (FFT based)

   – rebin option governs frequency space rebinning (0 disables; -ve for geometric series rebinning; +ve for constant rebin factor)

powspec cfile1=00434928000_out/sw00434928000xwtw2po_cl_bcorr_t0.0179s_orb1.lc window='-' dtnb=-1 nbint=4096 nintfm=INDEF rebin=0 plot=yes plotdev=/xs outfile='default'



rebin=0



rebin=-1.02

X-ray Timing Analysis - 4

- By default, powspec normalises the average powerspectrum following Leahy et al (1983) so that the Poisson noise powers follow a χ^2 distribution with 2MW d.o.f. scaled by 1/MW (M = number in average, W = rebinning factor) → average power level of 2.
  - Can be used to set detection thresholds (see Leahy et al 1983)
- Quasi Periodic Oscillations (QPOs) produce broad powerspectrum peaks → most sensitive results obtained when frequency binning matches QPO width.
- Coherent oscillations (e.g. pulsars) produce narrow peaks → most sensitive when long duration datasets are used (as frequency resolution 1 / T, where T is duration)

Alternate Methods:

- Epoch folding (efsearch) – can refine period by oversampling in frequency space
- Lomb-Scargle – can refine period by oversampling in frequency space
- Z_n^2 (applied to events directly)

X-ray Timing Analysis - 5

Beware

- Binning artefacts can be seen if light curve time bin size is close to a small integer multiple of the native readout time (TIMEDEL in event file) – can cause false signals in the powerspectrum
    - Try different time bin sizes to see if any proposed signals remain
- Pile-up affects the noise power levels & reduces the amplitude of detected signals
- Large data gaps, or unevenly sampled data (e.g. caused by Swift's orbit) - resulting powerspectrum is a convolution of signal with data window function
    - Lomb-Scargle periodogram can be less susceptible when searching for periods in unevenly sampled data

Resources

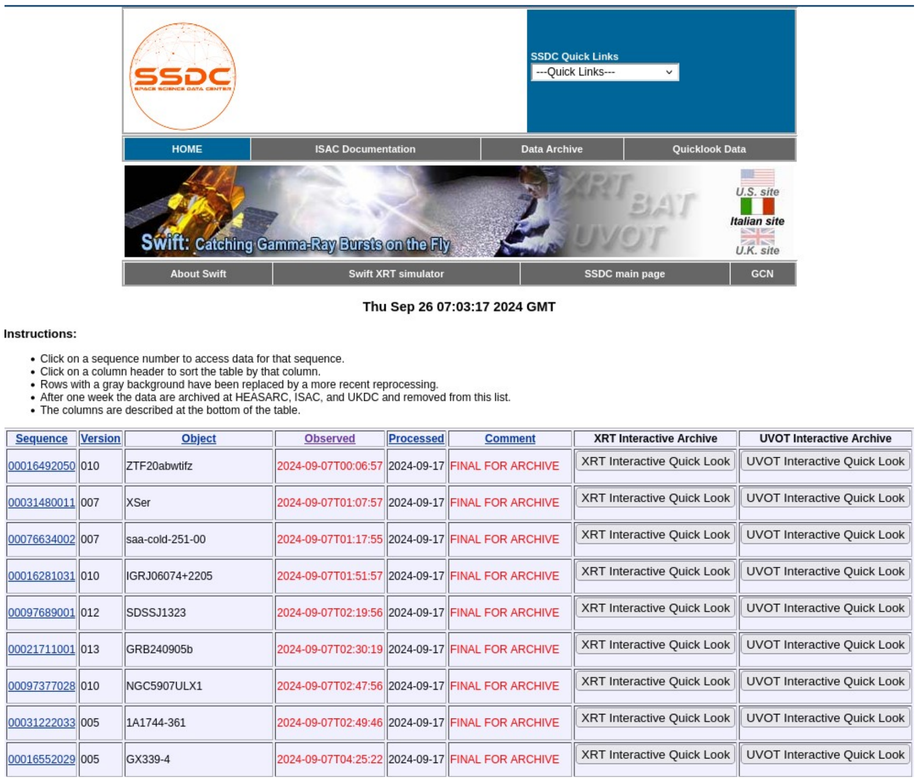
- Leahy et al. 1983, ApJ, 266, p. 160 and 272, p. 256 – PSD and Epoch folding
- van der Klis, M. 1989, "Fourier Techinques in X-ray Timing", in Timing Neutron Stars, NATO ASI 282
- Israel, G. – High Energy Astrophyiscs Summer School, 2008, Urbino – Basics of X-ray Timing (asd.gsfc.nasa.gov/urbino/presentation/Israel_Urbino08_t3.pdf)

Barrett & Vaughan 2012 ApJ 746 p. 131 – ML Fitting of PDS

Stingray – docs.stingray.science/en/stable/ – Comprehensive timing analysis package in python

Swift – UVOT Data Analysis

There is no direct UVOT equivalent of the XRT product generator, but the Italian SSDC provide links on their version of the Quick-Look site for online interactive UVOT image analysis.


SSDC Quick Links
---Quick Links---

HOME | ISAC Documentation | Data Archive | Quicklook Data

Swift: Catching Gamma-Ray Bursts on the Fly

U.S. site
Italian site
U.K. site

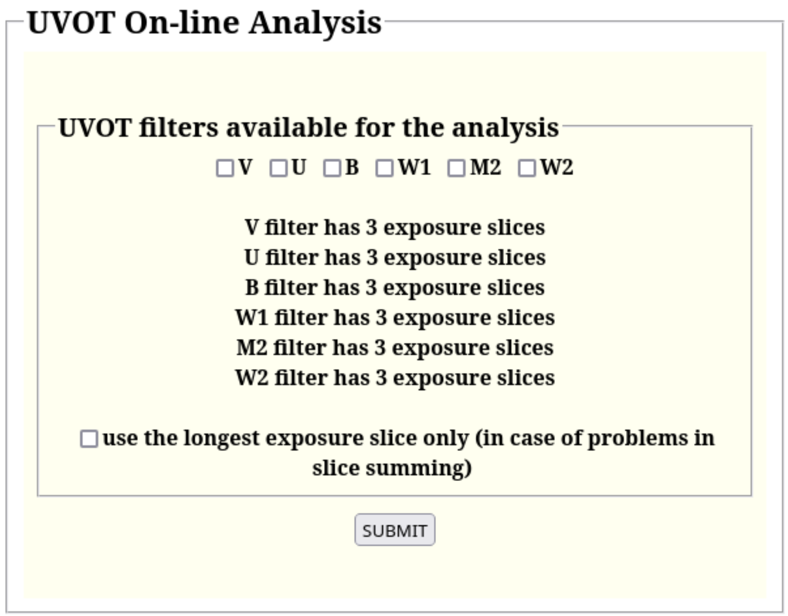
About Swift | Swift XRT simulator | SSDC main page | GCN

Thu Sep 26 07:03:17 2024 GMT

Instructions:

- Click on a sequence number to access data for that sequence.
- Click on a column header to sort the table by that column.
- Rows with a gray background have been replaced by a more recent reprocessing.
- After one week the data are archived at HEASARC, ISAC, and UKDC and removed from this list.
- The columns are described at the bottom of the table.

Sequence	Version	Object	Observed	Processed	Comment	XRT Interactive Archive	UVOT Interactive Archive
00016492050	010	ZTF20abwtifz	2024-09-07T00:06:57	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00031480011	007	XSer	2024-09-07T01:07:57	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00076634002	007	saa-cold-251-00	2024-09-07T01:17:55	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00016281031	010	IGRJ06074+2205	2024-09-07T01:51:57	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00097689001	012	SDSSJ1323	2024-09-07T02:19:56	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00021711001	013	GRB240905b	2024-09-07T02:30:19	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00097377028	010	NGC5907ULX1	2024-09-07T02:47:56	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00031222033	005	1A1744-361	2024-09-07T02:49:46	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look
00016552029	005	GX339-4	2024-09-07T04:25:22	2024-09-17	FINAL FOR ARCHIVE	XRT Interactive Quick Look	UVOT Interactive Quick Look



UVOT Data

UVOT data can be collected in Image, Event or Image&Event mode.

Image mode provides a 2-dimensional 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. Most ToO observations will be in Image mode, unless Event mode is explicitly requested.

File names: sw01310284000uuu_sk.img

u for UVOT

sk = sky image, in WCS coordinates; ex = exposure map, rw = raw – detector coordinates only.

11 digit ObsID, consisting of 8 digit target ID/trigger number and 3 digit segment number.

Two letters identifying the filter: wh(ite), vv, bb, uu, w1, m2, w2

Aspect Correction

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

> fkeyprint 01310284000/uvot/image/sw01310284000uuu_sk.img ASPCORR

If ASPCORR = DIRECT, the correction has been applied. If this is not the case, it **may** be possible to correct the data, but it will be fiddly! You could try a manual aspect correction using uvotunicorr.

The information about the star catalogue file required can be obtained from
https://swift.gsfc.nasa.gov/analysis/threads/uvot_thread_aspcorr.html
though may need to use http://archive.eso.org/skycat/servers/usnoa-server
for the location, rather than the Harvard server.

Note that, even if ASPCORR = DIRECT, there are some cases where the spacecraft is drifting, and point sources will appear elongated.

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 single file can be summed:

>uvotimsum

Name of input image file: sw01310284000uuu_sk.img(.gz)

Output file name: 000_usum.fits

or, if a list of images is provided (comma-separated, or a text file with a single image per line; this can be read in as @<filename>), every extension in the files will be combined:

>uvotimsum

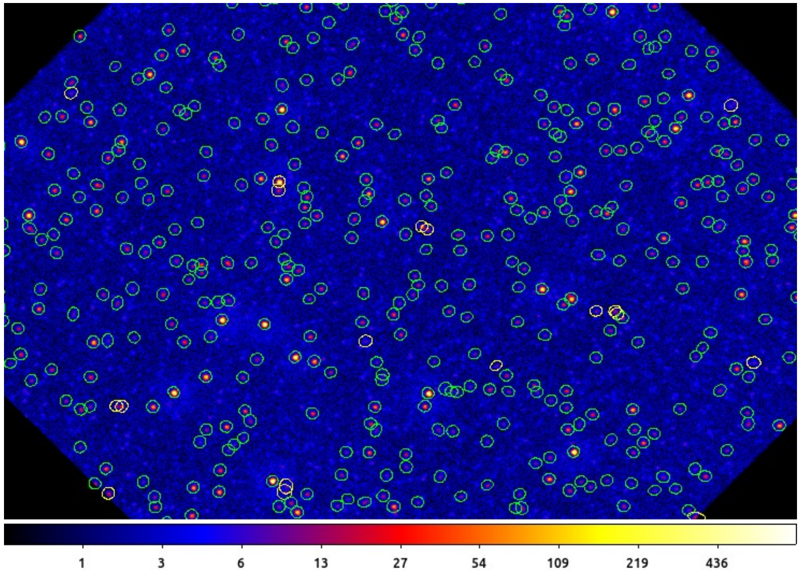
Name of input image file:

sw01310284000uuu_sk.img(.gz),sw01310284001uuu_sk.img(.gz)

Output file name: 000-001_usum.fits

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 expopt=BETA

Input image file: sw01310284000uuu_sk.img+1

Output file name: udetect.fits

Exposure map file name or NONE:

sw01310284000uuu_ex.img

Detection threshold (0.5:100): 3

udetect.fits lists positions, count-rates and 1σ errors for each source.

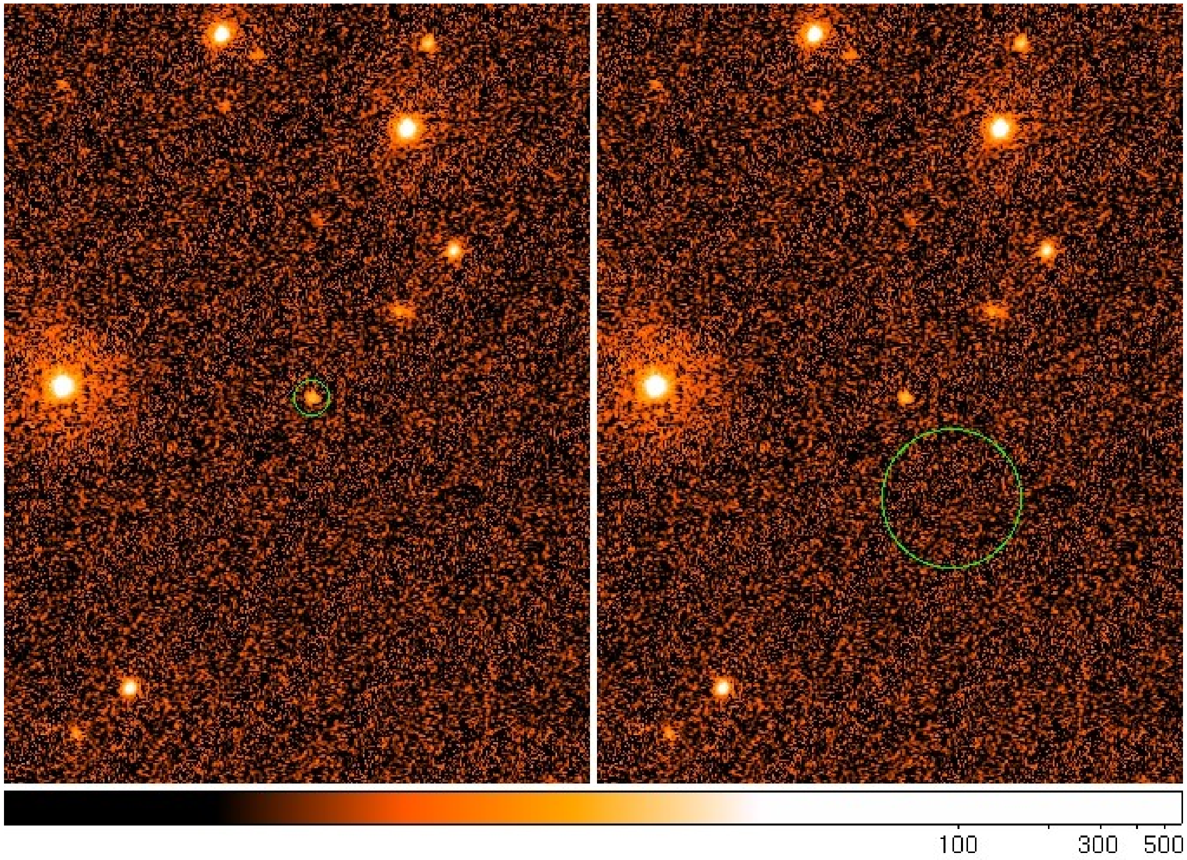
If a summed image is used with an exposure map, include expopt=ALPHA on the command line.

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. Better than relying on uvotdetect.

It is best to use a circle with a radius of 5 arcsec for the source, since this was how the calibration was done; a smaller region (eg 3 arcsec) can be used if there are nearby sources. 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.



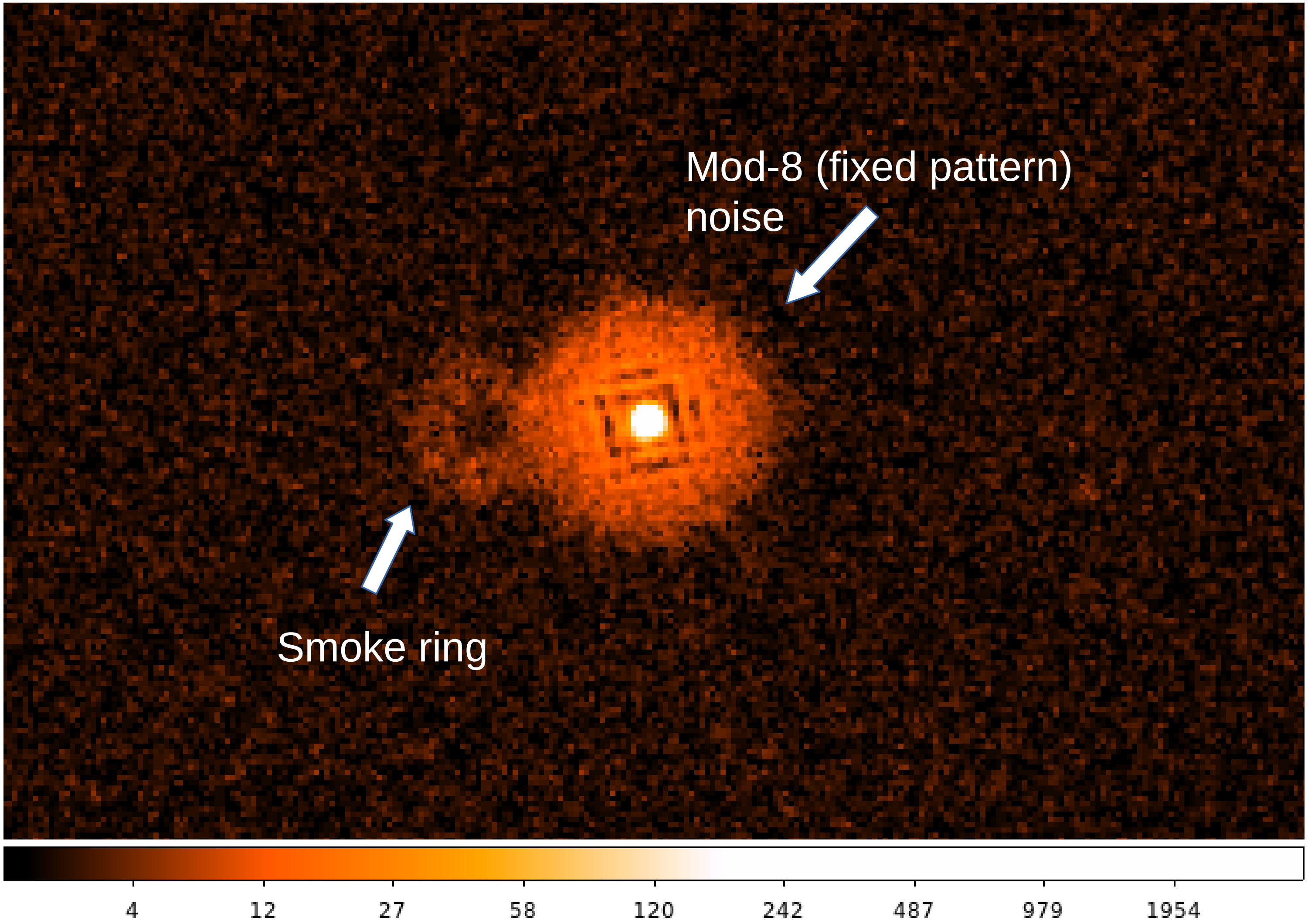
uvotsource - 2

> uvotsource
Input image file and extension[ ]: sw01310284000uuu_sk.img+1
Source region file[ ]: source.reg
Background region file[ ]: back.reg
Background threshold (1:100) [ ]: 3
FITS table of values[ ]: uphotometry.fits

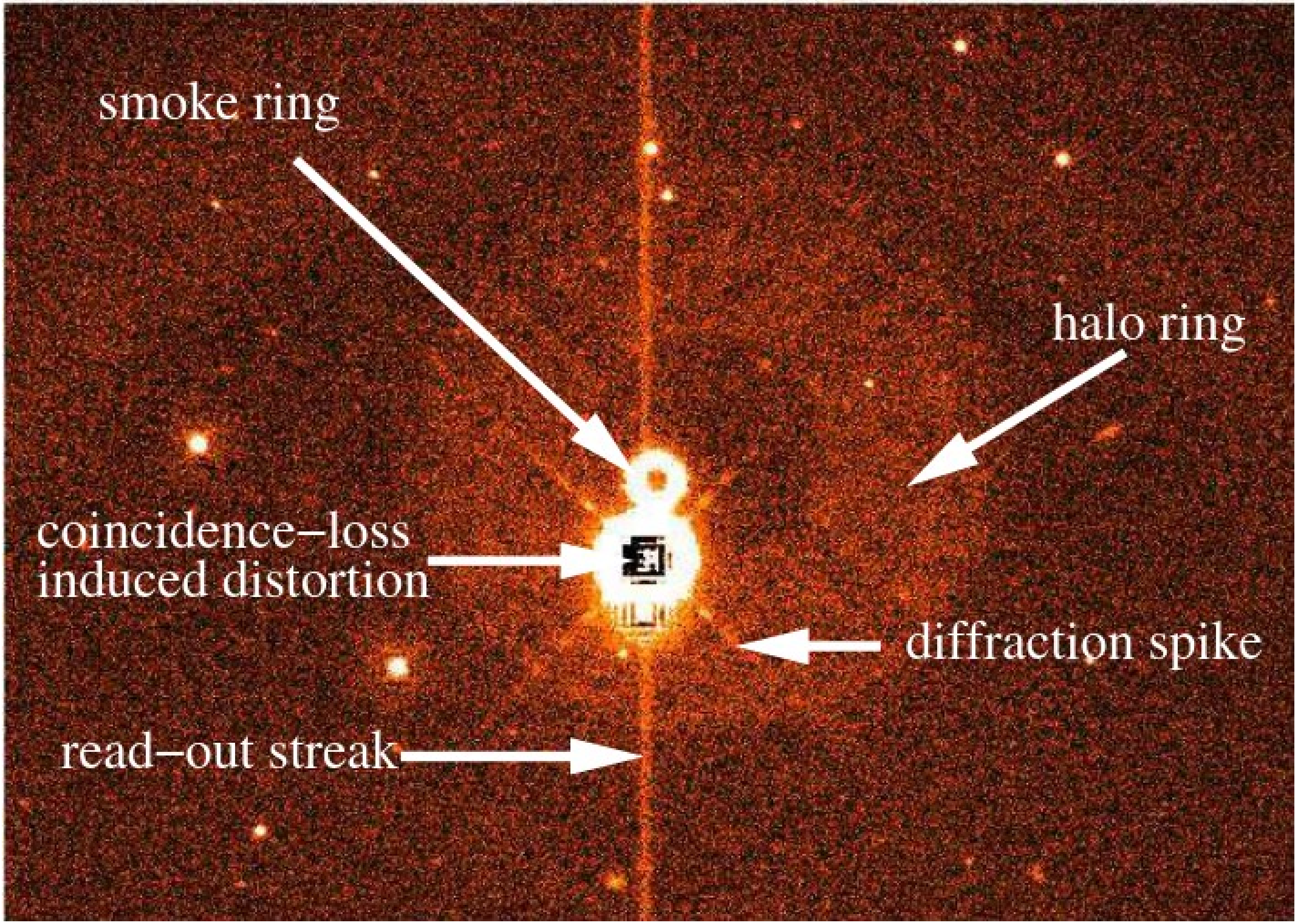
Results are then written both to the screen and the output FITS file, including detection significance, coincidence-corrected count rate, magnitude and flux density (or "Background threshold"-sigma upper limits if the source is undetected: 3σ in the above example).

There are areas of the UVOT CCD where the throughput is lower; if your source overlaps with one of these, the magnitude will be shown as 99. In the output file, CORR_RATE will also appear as -999 and SSS_FACTOR as -99. Fluxes are not marked as bad in the same way, though. (No idea why not!)
Note this won't happen for summed images – so best to check separate extensions.
See https://www.swift.ac.uk/analysis/uvot/sss.php for more details.

Coincidence loss



Artifacts



From M.J. Page et al. (2014) in Proc. of Science 233, “Swift: 10 Years of Discovery (SWIFT 10)”, ed. P. D'Avanzo, id.37

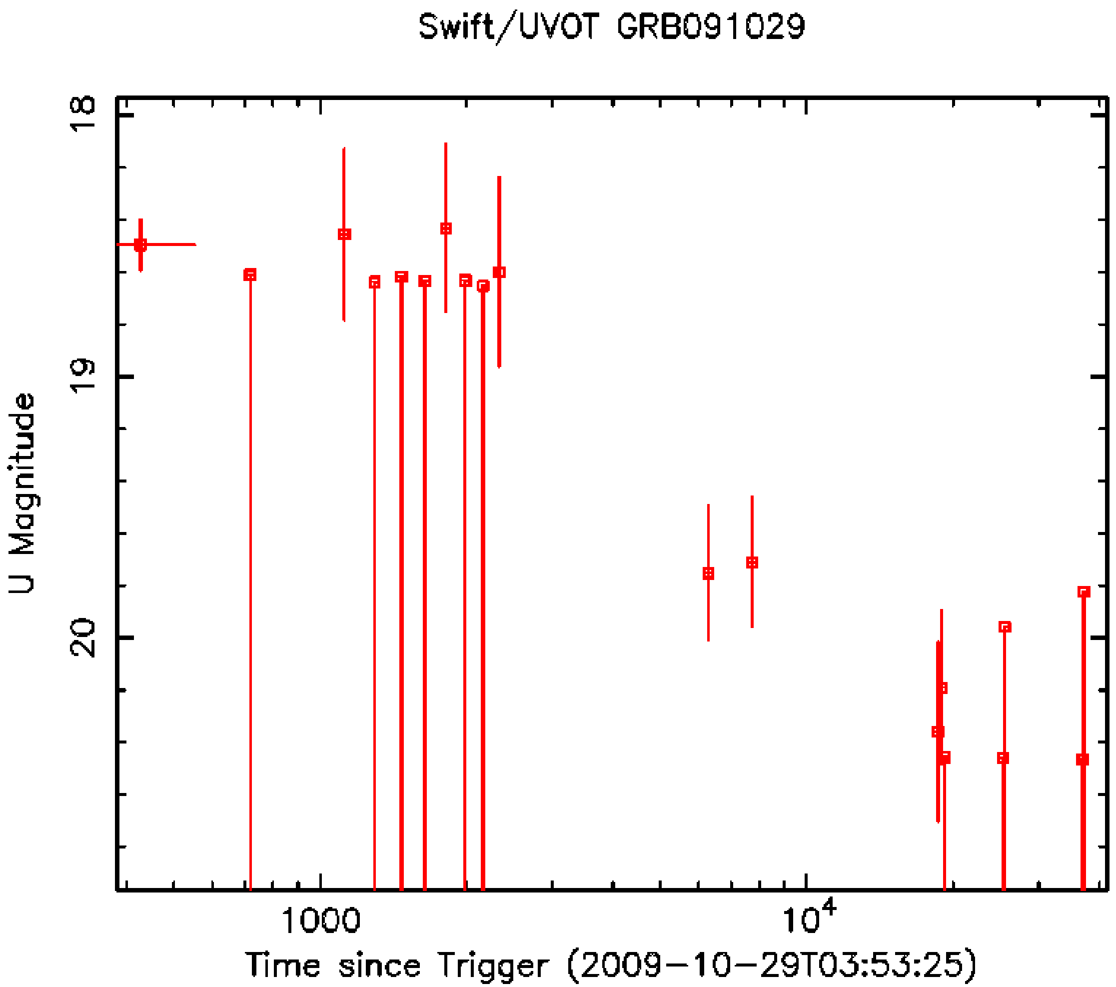
uvotmaghist -1

The uvotmaghist tool runs uvotsource on all the aspect-corrected extensions of the input image to create a light curve. A list of files can also be provided as the input using @filename.

> uvotmaghist
Name of input image file[ ]: sw01310284000uuu_sk.img
Name for output magnitude history[ ]: umaghist.fits
Name for output magnitude history plot or NONE[ ]: umaghist.gif
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 (though the default scaling may be off!). One point will be plotted for each extension of the image. If any of these are non-detections (at the 3σ 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



uvotproduct -1

Sometimes individual images will not provide detections, and data will need to be summed to get a sensible lc. uvotproduct combines images until significant detections are reached.

The input file(s) can be a single *sk.img(.gz) file, a comma separated list of images or @filename where filename is a file containing 1 sky image on each line. If the TRIGTIME keyword is not in the file headers, timezero=xxxxxxxxx.x should be included on the command line (Swift MET).

By default, points are merged until snr=2. To increase this to 3, include rebin=MIN_SIGMA=3,SNR_DIFF=3 on the command line (comma, no space).

The MAX_TIME_RATIO and MIN_TIME_RATIO control when data points will be merged. MAX_TIME_RATIO=2 (default) means bins with a ratio of new end time to start time > 2 will not be combined. The MIN_TIME_RATIO value means that even significant detections will be combined if they are very close in time.

uvotproduct -2

> uvotproduct
Name of input image file[ ]: @filename
Source region file [ ]: source.reg
Background region file [ ]: back.reg
Name for output report or NONE[ ]: report.txt
BAT position[ ]: NONE
XRT position[ ]: NONE
UVOT position[ ]: NONE
Ground position[ ]: NONE
Name for output magnitude history plot or NONE[ ]: uvotprod.gif
Name for output magnitude history [ ]: uvotprod.fits

If the background region file is set to DEFAULT, the background will be extracted from an annulus centred on the source (12.5-25 arcsec), with 6 arcsec exclusion regions for any sources detected within (or overlapping) the background annulus.

Note that uvotproduct does not take into account the SSS flagging when merging data points; it will include data which fall on the affected areas of the CCD by default.

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 20041120v106 (e.g. swuuu_20041120v105.rsp). Run uvot2pha, adding the extension number to the input filename.

```
> uvot2pha
Input (img or evt) filename[ ]: sw01310284000uuu_sk.img+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[ ]: CALDB
```

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=sw01310284000uuuw1po_uf.evt eventext=EVENTS teldef=CALDB attfile=sw01310284000sat.fits.gz aberration=n randomize=y seed=836 ra=269.6172 dec=-40.3661
```

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=sw01310284000uuuw1po_uf.evt attorbfile=sw01310284000ao.fits.gz outfile=sw01310284000uuuw1po_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 [ ]: sw01310284000uuuw1po_cl.evt
Output light curve filename [ ]: u.lc
Source region filename [ ]: src.reg
Background region filename [ ]: back.reg

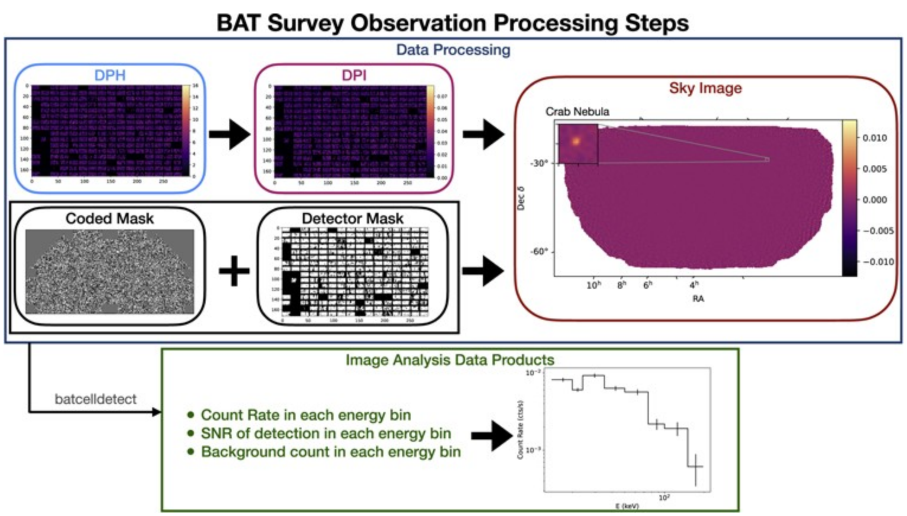
“timebinalg=g gtifile=gti” is the alternative, where gti is the relevant GTI file.

Swift – BAT Data Analysis

If BAT triggers on a gamma-ray source (typically a GRB, but sometimes other sources), then Swift saves event files and TDRSS data, and the standard batgrbproduct script can be used.

Most of the time, however, BAT operates in survey mode, collecting Detector Plane Histograms. There is now an API to analyse these data, called BatAnalysis, available at https://github.com/parsotat/BatAnalysis

See Parsotan et al. (2023, ApJ, 953, 15) for details.



Where to get help

We provide online data analysis threads at Leicester

https://www.swift.ac.uk/analysis/xrt/
https://www.swift.ac.uk/analysis/uvot/
https://www.swift.ac.uk/analysis/bat/

as well as an email helpdesk: swifthelp@leicester.ac.uk

Home > Data Analysis > XRT

XRT Data Analysis

Ready-made products

Swift-XRT data products for GRBs
Build Swift-XRT products for any object
Information about available XRT Positions for GRBs

XRT analysis threads

- General introduction
  - Obtaining and setting up the Swift software
  - Files and directory structure
- General processing
  - Producing cleaned XRT event-files
  - Source detection and position determination
- Analysis
  - General introduction to XSELECT
    - How to extract an image
    - How to extract a spectrum
      - Pile-up walk-through
      - ARFs
      - Position-dependent WT RMFs
    - How to extract a light-curve
      - Light-curve exposure correction
      - Barycentric correction
  - Exposure Maps

Home > Data Analysis > UVOT

UVOT Data Analysis

UVOT analysis threads

This page is designed to provide a starting point for the analysis of UVOT images. UVOT data are taken in 3 modes: Image mode, Event mode and Image&Event mode. There are 7 broadband filters and 2 grisms, and various window sizes. The subsequent pages deal with analysis of Image mode data. Event mode data are treated in a similar way to X-ray data. For analysis of Event mode and grism observations you are referred to the UVOT Software Guide, the UVOT Digest and Paul Kuin's "brief data reduction guide for UVOT grism data" (though see also limits for the grism). Online help for any of the UVOT tasks can be found with `fhelp` taskname or `plist` taskname.

Here we assume that at least the first batch of data has come down (after the TDRSS data), and we start from the Level 2 products. Steps to convert Level 1 into Level 2 products are not outlined here; for this you are referred to the Swift UVOT Software Guide. Particular care should be taken with the reduction and analysis if the field is crowded, the background is high or your source is very bright or very faint.

New DRESSCode is a python pipeline to reduce Swift UVOT images of extended sources written by Marjorie Decleir of STScI.

- General introduction
  - Obtaining and setting up the Swift software
  - Files and directory structure
- General processing
  - Position determination
- Analysis
  - Magnitude and flux determination
  - How to sum images
  - How to create a light-curve
  - How to create spectral files
  - uvotproduct

Home > Data Analysis > BAT > index

BAT Data Analysis

This page explains the steps required to obtain sky images, spectra and light-curves from BAT **event data**. Running these scripts will result in a BAT spectrum and response matrix, which can then be used in XSPEC, along with spectra from the XRT and UVOT. **If you are interested in analysing DPH/survey data, please see the paper by Parsotan et al. (2024). Please note that the UKSSDC team are unable to provide help with the BatAnalysis Python package.**

Many of these steps are performed by the pipeline at the SDC, but these individual scripts allow users to perform the data analysis "by hand". This is useful both for understanding what the processing does (rather than accepting the outputs of a "black box") and also if a problem with the pipeline were to develop.

The individual scripts are covered below, with some detail about the input files required for each step in the production of level 2 BAT data. Although the entire set of parameters can be typed directly at the command prompt, the required files do have to be in a specific order, which is not immediately obvious. However, the BAT ones will prompt for the required inputs, which simplifies matters. For example, typing `batbinevt` will then prompt for input event file name etc. There are, however, some 'hidden' settings, which will not be prompted for; these can be shown by using the FTOOL command `plist`, which will show all the parameters, including the hidden values, given in parentheses (example). The hidden defaults can be overridden in 1 of 3 ways: the new value can be defined on the command line (this is the method which has been used below, when going through the individual scripts), the `pset` command can be used or the `par` file can be edited. For more details on the `pfiles` commands, see the FTOOLS Tutorial.

There are many other BAT tasks which can be used to perform tasks such as refining the GRB position (`batcelldetect`) and cleaning the background (`batclean`). Running `fhelp swift` lists and explains these tools.

Note that HEASoft includes a complete GRB processing script, batgrbproduct, which generates all the standard products (images, light-curves and spectra) from the observations automatically. This is the recommended method to analyse BAT data. If time-sliced spectra or different energy bands for the light-curves are required, then the relevant threads listed below can be followed.

If a GRB is detected close to the edge of the BAT field of view (low partial coding fraction), the default processing may not show any counts during this time interval. To work around this, the command **`batgrbproduct aperture=CALDB:DETECTION pcodethresh=0.01 imgpcodethresh=0.01 extractor=fextract-events`** should be run.

Instrument guides

BAT data analysis guide
XRT data analysis guide
UVOT data analysis guide

BAT help threads

General introduction
- Obtaining and setting up the Swift software
- Files and directory structure

General processing
- Energy conversion
- Quality maps, hot pixels and mask weighting

Analysis
- Creating images
- Creating light-curves
  - Duration determination
- Creating spectra

BAT documents

BAT Software Guide
BAT instrument paper

BAT help

GSFC BAT Analysis Threads
BAT Digest Page at GSFC
Attend a Swift training session.

Helpdesk

Questions about Swift? Try our guide to Swift or the guide to data processing and analysis. If these don't solve your problem, please feel free to e-mail us at swifthelp@le.ac.uk.

List of acronyms and abbreviations

We are located in the Department of Physics & Astronomy, at the University of Leicester (directions).