BurstCube Analysis Guide

Version 1.0

DATE July 31 2025

Goddard Space Flight Center
Greenbelt, Maryland

Prepared by: James Runge and Lorella Angelini (HEASARC)

Table of Contents

1. OVERVIEW

The BurstCube analysis package within HEASoft has been developed based on the package **bc-tools**, https://gitlab.com/burstcube/bc-tools. Within HEASoft, the package contains five tasks: *bcrebevt*, *bctimebst*, *bcoccult*, *bcfindloc*, and *bcprod*. These tasks may be run in a variety of ways: a) from command-line similarly to any task within HEASoft, b) within Python using HEASoftpy, c) or from command-line without having HEASoft installed. The tools uses a Python module astro-gdt (https://astro-gdt.readthedocs.io/en/latest/) . If the software is run as option a) or b) the package atro-gdt is included in HEASoft , instead if run as case c) the astro-gdt Python module needs to be installed by the user.

This guide provides examples to how run the tasks in command-line within the HEASoft environment (case a). Example to how run the task in Python using HEASoftpy interface is available from the website https://heasarc.gsfc.nasa.gov/lheasoft/heasoftpy/.

These tasks were designed assuming to have in the archive an attitude file for each observation. The lack of attitude limits the usage of some of these tasks. This limitation is highlighted in the task description. The tasks are presented in the order in which they are intended to run as described by the flowchart.

2. TASKS

2.1 Rebinning the data (bcrebevt):

The first step to analyzing BurstCube data is reformatting the data so that it may be used by further analysis tools. For EVENT data, this entails binning the data into energy channels (either 16 or 64) as well as binning by time. Continuous binned data (CBD) is not rebinned as it is already in a 16-channel format by energy.

The task in charge of accomplishing this is called bcrebevt. Its usage is explained in the following paragraphs.

The main command-line usage of bcrebevt looks like the following:

```
bcrebevt infile outprefix timecol rebincol chanbin /
         (ebounds) (rebin) timebin (plotbin)
```

Description:

BurstCube collects the data in three different modes: The BurstCube science data are collected in three different modes: Trigger Time Tag Event (T3E or Requested Time Tag Event, RTTE), Alert Trigger Data (ATD) and Continuous Binned Data (CBD).

Data are collected on board by events. The events collected are binned on board and send on ground as CBD data where a channel spectra is calculated every NN seconds. If a trigger occurs the data are downloaded as T3E as well as the ATD. The FITS data format for CBD and ATD consists of a table with a TIME column, and a column contains an array with counts accumulated in 16 channels instead the T3E has in each row the time of the event and channel. The ADT includes time interval only around the burst and the FITS file contains several extensions with different time binned 0.064s, 0.256s, 1.024s, 4.096s. The CBD file contains one extension with a single binning time.

This task prepares the data with a common data format for following analysis by creating a FITS output file with three extensions. The first extension contains: a) a TIME column, b) a COUNTS column with array of counts in different channels (spectrum) and c) a SUMCOUNTS column with the sum of the counts in the array. The output file is equispaced in time and is referenced in this help and other BurstCube tasks as array. The other two extensions contain the Good Time Interval (GTI) and the Energy Boundary (EBOUND) information. If the input is an event file, the data are rebinned by channel (see parameter chanbin) and time (see parameter timebin). To rebin by energy it requires the CALDB files that contain the channel number and the associated energy (EBOUND) and the REBIN file that contains the channel minimum, maximum and the rebinning factor. For BurstCube there are only two possible spectral rebinning: 64 channels or 16 channels. If the input file is a CBD or ATD the data are with not rebinned neither by time or energy and the output is identical to the input with the column SUMCOUNTS added as well as the GTI and EBOUND extensions. If the input file is an ADT, the extension would need to be specified when input the else by default the task will use the first extension. The task also creates plots to show the lightcurve and spectrum for each detector and the data used in the plots are also written in a corresponding text file. The task requires to have the CALDB and REFDATA environment variables set, the first is used for the calibration file the second for the FITS template header of the output.

Parameters:

infile [file name]

Name of FITS files or @filename containing list of files (one per line). The extension may be specified within this parameter by adding to the filename +N or [N] where N is the extension number

outprefix [string]

Prefix to be used by all output files by this task.

timecol=TIME [string]

Name of the column containing the time information. By default, is set to TIME.

rebincol=PHA [string]
: Name of the column containing the information for the channel binning. By default, is set to PHA. The channel binning is only applicable to the EVENT file

chanbin=16 [int]
: Number of channels to rebin the EVENT data. Possible values are 16 or 64. The parameter does not apply to ADT and CBD which have counts already in 16 channels.

(*ebounds*=CALDB) [string]
: Name of the FITS calibration file containing the energy boundaries (file with the EBOUNDS extension) for each detector. By default, is set to CALDB and the file is read from the Calibration Database.

(*rebin*=CALDB) [string]
: Name of the FITS calibration file containing the rebinning scheme (file with the REBIN extension) for each detector. By default, is set to CALDB and the file is read from the Calibration Database. NOTE the rebinning ofr this input file needs to be consistent with the value in the parameter chanbin.

timebin=0.032 [double]
: Time bin width in seconds used to rebin the input EVENT data. This is not applicable if the input file contains CBD or ATD data

(*plotbin*=0.128) [double]
: Bin width in seconds used for lightcurve plots.

(*refdata*=REFDATA) [string]
: Directory name where the FITS template for the output is located. By default, this is set to environment variable REFDATA.

(*chatte*r=1) [int]
: Level of chatter/verbosity of the task. Allow value are between 0-5 default is 1.

(*log*=no) [bool]
: Flag to allow the log file creation. By default, the log file is not created (log=no), if the parameter log is set to 'yes' a log is created with a filename set to taskname_DDMMYYYY_HHMMSS.log.

(*clobber*=no) [bool]
: Flag to overwrite existing output files. By default, the file output is not overwritten (clobber=no)

(*mode* = ql) [string ql|hl|q]
: Mode to query the parameter file. Acceptable values include: 'ql' (query and learn/remember), 'hl' (hidden and learn/remember), 'q' (query but don't remember), 'h' (hidden).

Examples:

1. For EVENT files:
For rebinning EVENT files, the default values for timecol and rebincol are sufficient, but you may change chanbin to be 64 if wanted. As stated above, the input file can be a text file containing a list of files. The following is an example where all the EVENT files for a given observation (i.e. all detectors) are in a file called evt.lis:
evt.lis
./input/bc230117cs0_tte230117788_uf.evt
./input/bc230117cs1_tte230117788_uf.evt
./input/bc230117cs2_tte230117788_uf.evt
./input/bc230117cs3_tte230117788_uf.evt

Where the command-line would look like the following:
```
bcrebevt @input/evt.lis test TIME PHA 16 timebin=0.064
```
In this case, 16 energy channels and a time bin width of 0.064 is used.

2. For CBD files:
Since CBD files are already binned into 16 energy channels, the chanbin and timebin parameters are ignored. However, the rebincol parameter should be set to COUNTS. Example:
```
bcrebevt @cbd_1.txt cbd_1 TIME COUNTS 16 timebin=0.064
```

Outputs:

The bcrebevt task has multiple outputs:
outprefix*phaii.fits – An output FITS file for each input file
    File Structure:
    ARRAY_PHA: TIME, COUNTS (binned counts), SUMCOUNTS (sum of all counts)
    EBOUNDS: CHANNEL, E_MIN, E_MAX
    STDGTI: START, STOP
outprefix*binned_spectra.png – An image containing a plot of the energy spectra for all input files. A single file. See Fig.1(a)
outprefix*spec.txt – A text file containing the data used to create the spectrum plot. One per input file.
outprefix*lightcurves.png – An image containing a plot of the light curves for all input files. A single file. See Fig.1(b)
outprefix*lc.txt – A text file containing the data used to create the light curve plot. One per input file.

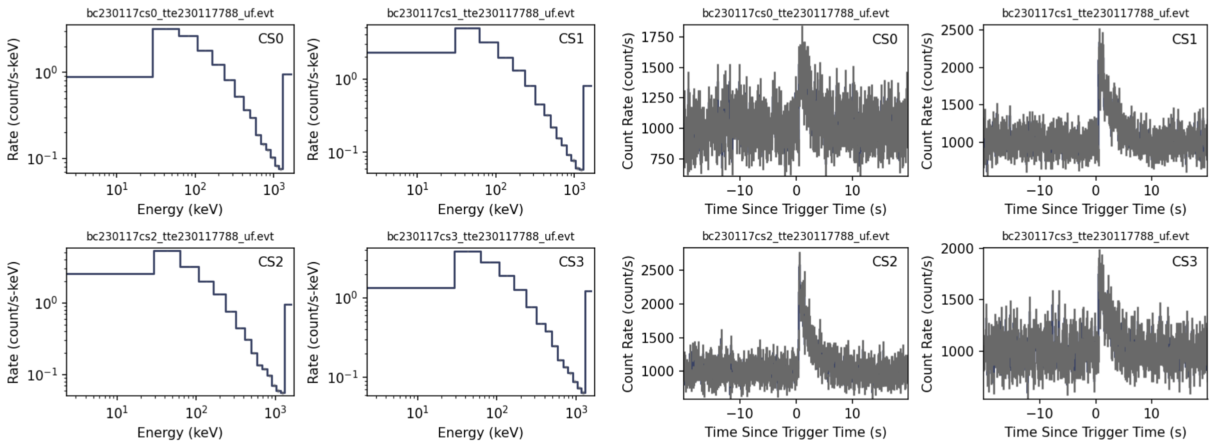


Figure 1: (a) On the left is the output image from bcrebevt showing the spectra for the input files of Example 1. (b) On the right is the output image from bcrebevt showing the lightcurves for the input files of Example 1.

2.2 Obtaining peak time interval (bctimebst):

Once the data has been reprocessed to the appropriate format, it is now time to identify the time interval of the burst within the data. This is accomplished using Bayesian block analysis of the lightcurves in the task *bctimebst*.

The command-line usage of the task is the following:

```
bctimebst infile outprefix timecol cntscol (emin) (emax)
```

Description:

This task calculates the time interval of the burst and additional intervals to characterize the burst as well as the background interval. The input data are the files output from *'bcrebevt'* and works for single detector or multiple detectors where each detector is in a separate input file. It is required for the input files to have the same time and channel binning, e.g. the TIMEDEL and TFORMS2 keywords values must be the same in all input files.

The task first calculates total lightcurve using the counts over the energy range specified by the parameter 'emin' and 'emax'. If data are from multiple detector (one detector per input file) the task combines data from all detectors over the energy range with the time binning value found in the TIMEDEL keyword of the input files. The total combined lightcurve is than rebinned using the Bayesian block algorithm (see Scargle et al., 2013, ApJ 764 167) and it is the baseline for the burst time interval calculation. The time interval is found with a three steps iterative process using the Bayesian block "lightcurves": a) First: Find the peak of the lightcurve and an estimate of the start and stop of the burst. b) Second: Fit the background using a polynomial fit in the lightcurve regions outside the estimated start and stop interval of the burst. c) Third: Subtract the fitted background from the lightcurve and using the latter lightcurve and restart for a) recalculating the peak and the burst interval.

The iterative process stops when two consecutive iterations provide the same time interval for the burst. At the first iteration the total lightcurve is used for b); for subsequent iteration the background regions are those outside of the first Bayesian block above background and the last Bayesian block above background. The time interval found within this process is the total time on burst to include the start and the stop of the burst.

The task also provides the T90 and T50 which are respectively the interval containing the 90% and 50% of the burst flux and the peak interval. A cumulative counts curve is derived from the total combined lightcurve from the total time interval on burst and the T90 is defined to include the 5% to 95% interval while the T50 includes the 25% to 75% interval. The peak interval is defined as the time interval with the highest number of counts in the combined lightcurve within the total time interval.

'bctimebst' outputs the times as good time intervals in a FITS file containing 5 extensions:

1- total time interval of the burst where the start and stop are defined as the 1st Bayesian block and last Bayesian block above background respectively

2- T90 defined as the 90% of the flux with the total time interval

3- T50 defined as the 50% of the flux with the total time interval

4- Peak defined as the time interval with the highest number of counts in the combined lightcurve within the total time interval

5- Background time interval determined within this task.

The task also outputs: a) plot file of the combined lightcurve (sum of all detectors) with the Bayesian block, the start and stop of the total burst, and the background level overplotted; b) Lightcurve of the combined data as FITS file and ASCII file; c) Bayesian block as FITS files.

Parameters:

infile [file name]

Name of FITS files or @filename containing list of files (one per line). The inputs appropriate for this task are the outputs of the 'bcrebevt'.

outprefix [string]

Prefix to be used by all output files by this task.

timecol=TIME [string]

Name of the column containing the time information. By default, is set to TIME.

cntscol=COUNTS [string]

Name of the column containing the information the information of the counts. By default, is set to COUNTS.

emin=45.3 [float]

Minimum energy in keV to consider.

emax=316. [float]

Maximum energy in keV to consider.

(*fprob*=0.05) [float]

This is the false alarm probability used in Bayesian block analysis for the the background fitting. The value ranges between 0 and 1. By default is set to 0.05 and not recommended to change.

(*poly*=3) [int]

The order of the polynomial used to fit the background. The default is set to 3, a compromise to not overfitting the data and avoid edge effects. Not recommended to change.

(*niter*=100) [int]

This parameter sets the maximum number of iterations for the background fitting. The value is set to a large number while testing shows that will converge quickly if the background is well modeled by the polynomial

(*bnum*=5) [int]

This value is a multiplicative number applied to the blocks before and after the burst to exclude leading and trailing edge intervals of burst. This parameter allows to estimate the background far away from the start and stop of the burst interval.

(*refdata*=REFDATA) [string]

Directory name where the FITS template for the output is located. By default, this is set to environment variable REFDATA.

(*chatter*=1) [int]

Level of chatter/verbosity of the task. Allow value are between 0-5 default is 1.

(*log*=no) [bool]

Flag to allow the log file creation. By default, the log file is not created (log=no), if the parameter log is set to 'yes' a log is created with a filename set to taskname_DDMMYYYY_HHMMSS.log.

(*clobber*=no) [bool]

Flag to overwrite existing output files. By default, the file output is not overwritten (clobber=no)

Examples:

Running *bctimebst* using a list of files in phaii.lis using the default energy range. The contents of phaii.lis are the files from *bcrebevt*:
./input/test_cs0_evt_230306702_phaii.fits
./input/test_cs1_evt_230306702_phaii.fits
./input/test_cs2_evt_230306702_phaii.fits
./input/test_cs3_evt_230306702_phaii.fits

```
bctimebst infile=@phaii.lis outprefix=test timecol=TIME /
cntscol=COUNTS
```

Note: If using CBD data as input, be sure to trim the input data to a smaller time window of interest. This is due to the fact that CBD data is an entire day's worth of data, not just the data around a single burst. Trimming the data will ensure that the time intervals are reasonable for the burst you are interested in.

Output:

The following files are the output from *bctimebst*:
outprefix.gti – GTI file with five extensions containing the time intervals for the following: TOTAL (entire time of the burst), T90 (time interval containing 90% of total flux of burst), T50 (time interval containing 50% of total flux of burst), PEAK (time interval of highest point of burst), and BKG (time intervals of background).
outprefix_bblock.fits – FITS file with table containing the following information of the Bayesian block analysis: START time of Bayesian block, STOP time of Bayesian block, COUNTS within the block, EXPOSURE time of block (STOP - START), and the block index, BINDEX.
outprefix_burst.lc – FITS file with a table containing information about the lightcurve with the following columns:

TIME – Time of bin
RATE – Background subtracted rate in cts/s
ERROR – Estimated error for rate in cts/s
TOTRATE – Non-background subtracted rate in cts/s
TOTRATE_ERROR – Estimated error for TOTRATE in cts/s
BKGRATE – Background rate in cts/s
BKG_ERROR – Estimated error of BKGRATE in cts/s

outprefix_total_lc.txt – CSV file containing the columns TIME, TOTRATE, and BKGRATE from outprefix_burst.lc.
outprefix_signal_id.png – An image containing a plot of the lightcurve showing the Bayesian blocks, the estimated background, and the START/STOP of the burst interval. See Fig. 2 below.

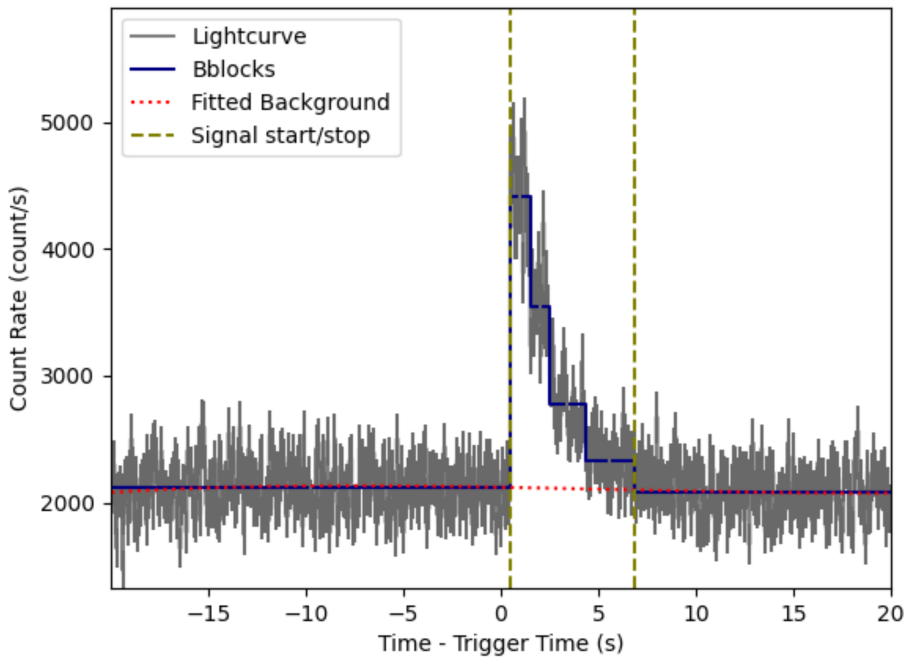


Figure 2: Output image from bctimebst. The lightcurve is shown in gray, the Bayesian blocks are in blue, the fitted background is dotted red, and the start and stop of the burst interval is shown in dashed, dark yellow.

2.3 Creating occultation map (bcoccult):

To account for the occultation due to the Earth, an occultation map must be made to mask parts of the data. This is accomplished using the task *bcoccult.*

Usage:

```
bcoccult orbit_file gti_file outprefix (nside) (scheme)
```

Description:

This task calculates the BURSTCUBE occultation full-sky map that shows the part of the sky occulted by the earth for a specific time period. The output map is written as a FITS file with an Aitoff projection in the primary header and a HEALPix representation in the first extension. The pixel values are set either 0 to indicate earth occultation or 1 to indicate no occultation. The task requires as input an orbit file and a good time intervals over which the earth occultation is evaluated. Both files are in FITS format. For each time of the orbital data file that fall within the input time interval, the task retrieves the satellite position and calculates a radius that represents the apparent size of the earth on the sky and sets to 0 pixels in the output image to indicate that the sky is occulted otherwise set the pixels to 1.

The task requires to have the REFDATA environment variable set to get the FITS template header of the output.

Parameters:

orbit_file [file name]
    Input FITS file containing orbital data. The columns read: TIME, X, Y, and Z.
outprefix [string]
    Prefix to be used by all output files by this task.
gti_file [file name]
    Input Good time Interval (GTI) FITS file containing time period to create occultation map.
(*nside*) [int]
    Size of HEALPix map to create. Must be power of 2. The default is 16
(*scheme*='ring') [string]
    Ordering scheme for HEALPix map. Either 'ring' or 'nested'
(*refdata*=REFDATA) [string]
    Directory name where the FITS template for the output is located. By default, this is set to environment variable REFDATA.

(*chatter*=1) [int]

Level of chatter/verbosity of the task. Allow value are between 0-5 default is 1.

(*log*=no) [bool]

Flag to allow the log file creation. By default, the log file is not created (log=no), if the parameter log is set to 'yes' a log is created with a filename set to taskname_DDMMYYYY_HHMMSS.log.

(*clobber*=no) [bool]

Flag to overwrite existing output files. By default, the file output is not overwritten (clobber=no)

(*mode* = ql) [string ql|hl|q]

Mode to query the parameter file. Acceptable values include: 'ql' (query and learn/remember), 'hl' (hidden and learn/remember), 'q' (query but don't remember), 'h' (hidden).

2.4 Localization map (bcfindloc):

Once the occultation map has been created, it is now time to find the location of the burst and create a localization map. This is accomplished using the task *bcfindloc.*
Due to lack of attitude in the BurstCube archival data , the following task cannot be used for all data but is described for completeness.

Usage:

```
bcfindloc infile outprefix burstgti attitude occultation
emin emax
```

Description:

This task calculates the likelihood position of a source and outputs a file with a map in the primary header in AIT projection header and a first extension with HEALPix representation. The values in the map represent the part of the sky occulted by the earth (pixel set to 0) or not occulted pixel set =1 and the pixels with the probably position have values between 0-1 (probability). It uses as input the occultation map calculated by *'bcoccult'*, the attitude file relevant to the observation and the science array file output of *'bcrebevt'*, the gti file the output of *'bctimebst'* and the calibration file containing the counts simulated for different models at different sky location.

'bcfindloc' reads the simulated calibration file, which contains theta, and phi and a HEALPix value together with rates, and use the quaternions in the attitude within the gti to calculate new theta and phi appropriate to the observation. The rates corresponding to the new theta and phi and the near 4 pixels are weighted average and the results places in pixel number of a HEALPix map. This step is performed for each quaternion included in the burst interval and the maps for each quaternion are multiplied by the duration of the quaternion. All maps are summed together so that each pixel contains counts. This procedure is done for each detector.

The log-likelihood is calculated for each pixel using the source and background counts, with the specific energy range, from the 4 detectors and the expected counts from CALDB adjusted for the attitude for each of the HEALPix of the map. The likelihood map is multiple by the occultation map and the higher value within this map is the "best position location" of the event for that spectral simulated model.

If there are multiple models in the simulated CALDB file, the task chooses the best model for the output map but also save the maps for the different models in a file with prefix "temp". The task needs to have set the location of the REFDATA directory that contains the headers of the output file and the CALDB location to access the calibration data.

Parameters:

infile [file name]
    Name of FITS files or @filename containing list of files (one per line). The file inputs for this task are the array files as generated by 'bcrebevt'
outprefix [string]
    Prefix to be used by all output files by this task.
burstgti [file name]
    Name of the FITS good time interval file containing time interval for burst. The file input for this task is the output of 'bctimebst'
attitude [file name]
    Name of the FITS attitude file.
occultation [file name]
    Name of the FITS Occultation file. This file input for this task is output of 'bcoccult'.
emin [float]
    Minimum energy in keV to consider.
emax [float]
    Maximum energy in keV to consider.
(*simloc*='CALDB') [string]
    Name of the FITS file containing the simulated counts for a given model for each detector. Default will search CALDB
(*extension*='all') [string]
    The simulated counts file contains several models laid out one per extension. This parameter allows to specify the extension to use for specific models either comma-separated list (1,2,3...) or 'all' to use all extensions.
(*poly*=3) [int]
    The order of the polynomial used to fit the background. The default is set to 3, a compromise to not overfitting the data and avoid edge effects. Not recommended to change.
(*refdata*=REFDATA) [string]
    Directory name where the FITS template for the output is located. By default, this is set to environment variable REFDATA.
(*chatter*=1) [int]
    Level of chatter/verbosity of the task. Allow value are between 0-5 default is 1.
(*log*=no) [bool]

Flag to allow the log file creation. By default, the log file is not created (log=no), if the parameter log is set to 'yes' a log is created with a filename set to taskname_DDMMYYYY_HHMMSS.log.

(*clobber*=no) [bool]

Flag to overwrite existing output files. By default, the file output is not overwritten (clobber=no)

(*mode* = ql) [string ql|hl|q]

Mode to query the parameter file. Acceptable values include: 'ql' (query and learn/remember), 'hl' (hidden and learn/remember), 'q' (query but don't remember), 'h' (hidden).

2.5 Creating final data products (bcprod):

Now that the time interval of the burst and the background has been identified, the final data products, which include source and background spectra, may be created. Using the outputs from the previous tasks (*bcrebevt* and *bctimebst*) as inputs, the task *bcprod* creates lightcurves and spectra for a variety energy ranges as well as time binning. Its usage is described below:

```
bcprod infile outprefix burstgti (erange1) (erange2) /
        (erange3) (timebin1) (timebin2) (poly)
```

Description:

This task takes as input files a list of array files (one file per detector) as created by *bcrebevt* and the GTI created from bctimebst and calculates lightcurve files, spectral files and plots of the products. *'bcprod'* reads the total time interval on burst and on background extensions from the GTI file and uses these time periods to calculate the spectra. For the source spectra (one for each detector) it adds all counts per channels within the total time on burst. For the background it fits the data with a polynomial of the order specified by the parameter poly. The fitting is performed channel by channel and background values are interpolated for the burst time region based on background fit. The background spectra (one per detectors) are output as results of this procedure and background value for the lightcurves are stored with the lightcurve output. From the input parameter 'radec', *'bcprod'* queries CALDB to retrieve the appropriate response file and write the filename in the source spectral file in the keyword "RESPFILE". The spectral files output are: a) source and background as type I format where there is one spectrum for the entire time interval, b) source and background as type II format where there is one spectrum for each timebin as for the original array input data file.

The lightcurves are extracted in three energy bands (see parameters 'erange1', 'erange2', 'erange3') and the total band with two different time binning (see parameters 'timebin1' and 'timebin2') for each of the detectors. The layout of the lightcurve for each detector contains eight extensions: four are using the 'timebin1' where each has data for a specific energy band and similarly the other four with 'timebin2'. Each extension contains the background subtracted rate and error (RATE and ERROR columns), the not background subtracted rate and error (TOTRATE and TOTRATE_ERROR columns) and the background rate and error (BKGRATE and BKGRATE columns)

'bcprod' also creates plots to show the lightcurves in all the available bands for each of the detectors (4 files) and the spectra for each detector together with the background spectrum. The task requires to have the CALDB and REFDATA environment variables

set, the first is used for the calibration file the second for the FITS template header of the output.

Parameters:

infile [file name]
Name of FITS files or @filename containing list of files (one per line).

outprefix [string]
Prefix for all output files.

burstgti [file name]
GTI file containing the source and background times for detected burst. (Output from bctimebst)

(*radec*=NONE) [string]
Right Ascension and Declination in the form "ra,dec".

erange1='50,100' [string]
First energy range in the format 'low,high'. Units: keV

erange2='100,300' [string]
Second energy range in the format 'low,high'. Units: keV

erange3='300,1000' [string]
Third energy range in the format 'low,high'. Units: keV

timebin1=0.064 [double]
First binwidth to do rebinning by. Units: seconds

timebin2=1.0 [double]
Second binwidth to do rebinning by. Units: seconds

(*poly*=3) [int]
Order of polynomial for background fitting.

(*refdata*=REFDATA) [string]
Directory name where the FITS template for the output is located. By default, this is set to environment variable REFDATA.

(*chatter*=1) [int]
Level of chatter/verbosity of the task. Allow value are between 0-5 default is 1.

(*log*=no) [bool]
Flag to allow the log file creation. By default, the log file is not created (log=no), if the parameter log is set to 'yes' a log is created with a filename set to taskname_DDMMYYYY_HHMMSS.log.

(*clobber*=no) [bool]
Flag to overwrite existing output files. By default, the file output is not overwritten (clobber=no)

(*mode* = ql) [string ql|hl|q]

Mode to query the parameter file. Acceptable values include: 'ql' (query and learn/remember), 'hl' (hidden and learn/remember), 'q' (query but don't remember), 'h' (hidden).

Example:

Running *bcprod* using a list of files within the text file phaii.lis (output from *bcrebevt*), setting the output prefix to be 'test' for all files, using the GTI file test.gti (output from *bctimebst*), and using the default values for the other parameters.

```
bcprod infile=@phaii.lis outprefix=test burstgti=test.gti
```

Outputs:

The final outputs from this task are as follows:
outprefixcs#_type.lc – A multi-extension FITS file with eight extensions, one for each combination of energy range (erange1, erange2, erange3, and total) and timebinning (timebin1 and timebin2). One file per detector (cs0, cs1, cs2, or cs3). Each extension has the following columns:

TIME – Time of bin
RATE – Background subtracted rate in cts/s
ERROR – Estimated error for rate in cts/s
TOTRATE – Non-background subtracted rate in cts/s
TOTRATE_ERROR – Estimated error for TOTRATE in cts/s
BKGRATE – Background rate in cts/s
BKG_ERROR – Estimated error of BKGRATE in cts/s

outprefixcs#_typelc.png – A figure containing the lightcurve plot for each combination of energy range and timebinning. The time interval of the burst is highlighted, and the fitted background is shown. See Fig. 3 for an example.
outprefixcs#_type.pha – A TYPE I PHA file for the burst (i.e. source). One file for each detector. Contains the following columns:

CHANNEL – Energy channel number
COUNTS – Number of counts in channel
QUALITY – Data quality

outprefixcs#_type_bg.pha - A TYPE I PHA file for the estimated background during the burst. Same structure as above.
outprefixcs#_type.phaii – A TYPE II PHA file for the burst (i.e. source). One file per detector. Contains the following columns:

SPEC_NUM – Spectrum number
CHANNELS – Array containing energy channel numbers
COUNTS – Array containing the number of counts in each energy channel
EXPOSURE – Exposure time for this time bin
BACKFILE – Associated background file
QUALITY – Quality of data

TIME – Start of time bin

ENDTIME – Stop if time bin

outprefixcs#_type_bg.phaii – A TYPE II PHA file for the estimated background during the burst. One file per detector. Same structure as above.

outprefixcs#_typepha.png – A figure containing a plot of the source spectrum and estimated background spectrum during time of the burst. See Fig. 4 for an example.

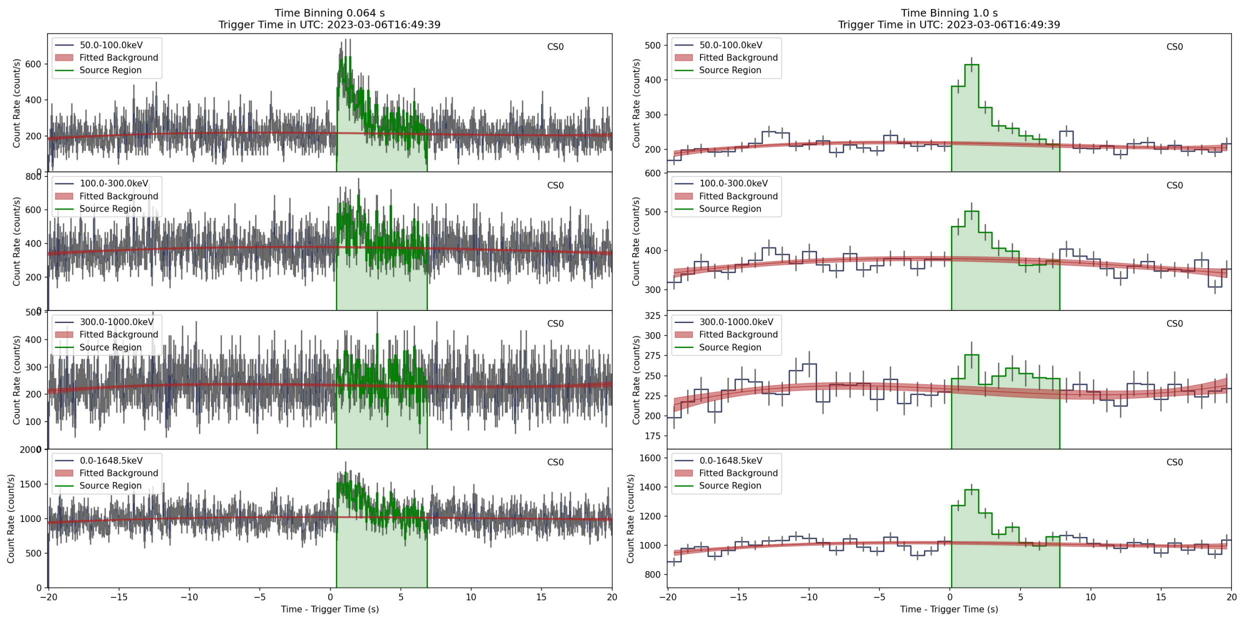


Figure 3: Output image from bcprod containing the lightcurves for a single detector. Left: Lightcurve for timebinning 1. Right: Lightcurve for timebinning 2. There is a lightcurve for each energy range as well as the total energy range. The fitted background and error are shown in red. The time interval of the burst is highlighted in green.

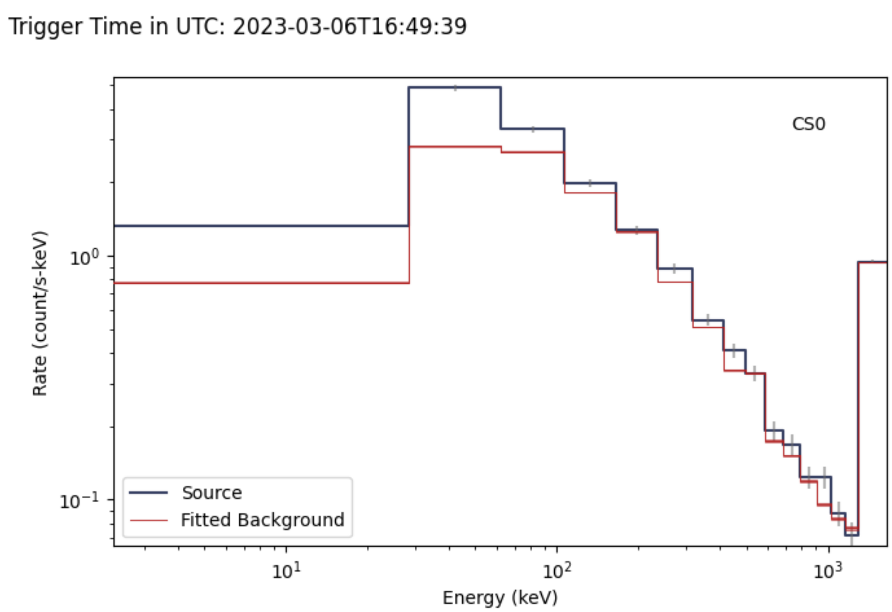
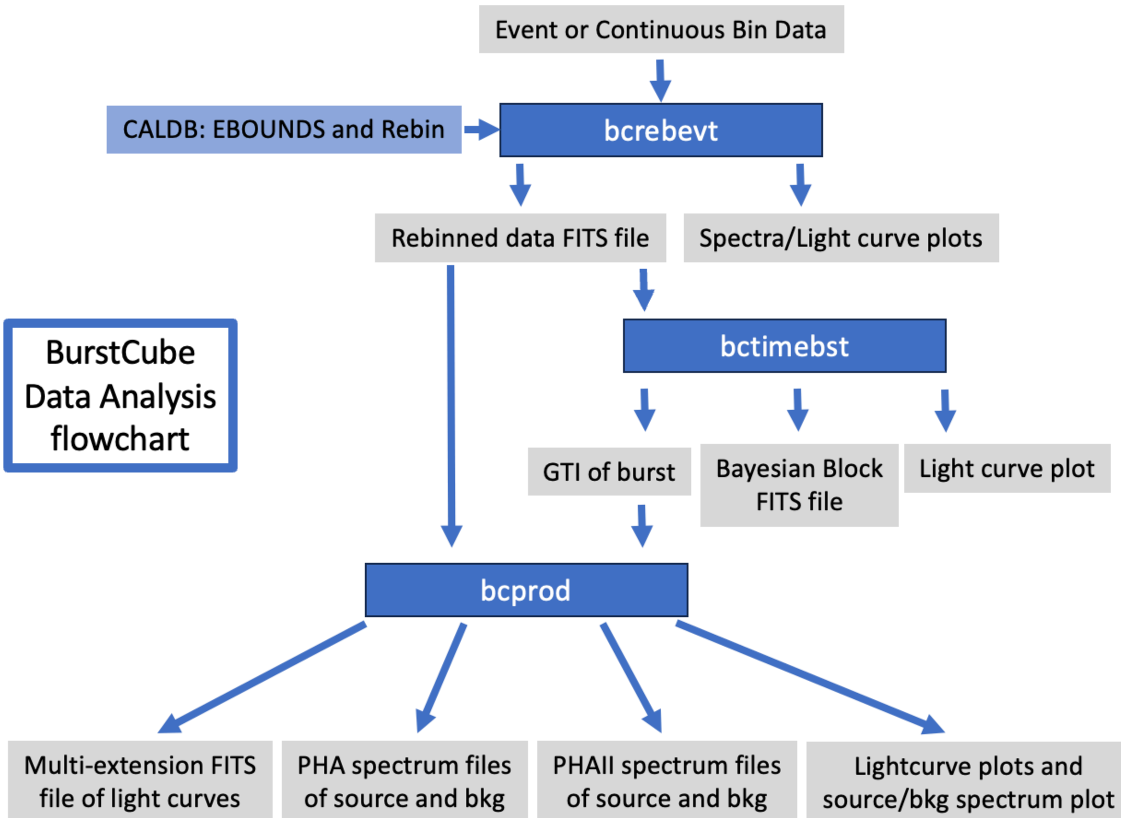


Figure 4: Output image from bcprod containing the spectra for a single detector. The source is in black and the fitted background is in red.

3. Flowchart


Event or Continuous Bin Data
CALDB: EBOUNDS and Rebin
bcrebevt
Rebinned data FITS file
Spectra/Light curve plots
BurstCube Data Analysis flowchart
bctimebst
GTI of burst
Bayesian Block FITS file
Light curve plot
bcprod
Multi-extension FITS file of light curves
PHA spectrum files of source and bkg
PHAII spectrum files of source and bkg
Lightcurve plots and source/bkg spectrum plot