Draft version June 10, 2025
Typeset using LaTeX **twocolumn** style in AASTeX63

BatAnalysis - A Comprehensive Python Pipeline for Swift BAT Time-Tagged Event Data Analysis

Tyler Parsotan,[1] David M. Palmer,[2,3] Samuele Ronchini,[4] James Delaunay,[4] Aaron Tohuvavohu,[5] Sibasish Laha,[6,1,7] Amy Lien,[8] S. Bradley Cenko,[1,9] Hans Krimm,[10] and Craig Markwardt[1]

[1] *Astrophysics Science Division, NASA Goddard Space Flight Center,Greenbelt, MD 20771, USA.*
[2] *Los Alamos National Laboratory, Los Alamos, NM 87544, USA*
[3] *New Mexico Consortium, Los Alamos, NM, 87544, USA*
[4] *Department of Astronomy and Astrophysics, The Pennsylvania State University, 525 Davey Lab, University Park, PA 16802, USA*
[5] *Cahill Center for Astronomy and Astrophysics, California Institute of Technology, Pasadena, CA 91125, USA*
[6] *Center for Space Science and Technology, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA.*
[7] *Center for Research and Exploration in Space Science and Technology, NASA/GSFC, Greenbelt, Maryland 20771, USA*
[8] *University of Tampa, Department of Chemistry, Biochemistry, and Physics, 401 W. Kennedy Blvd, Tampa, FL 33606, USA*
[9] *Joint Space-Science Institute, University of Maryland, College Park, MD 20742, USA*
[10] *National Science Foundation 2415 Eisenhower Ave., Alexandria, VA 22314*

Submitted to ApJ

ABSTRACT

The Swift Burst Alert Telescope (BAT) is a coded aperture gamma-ray instrument with a large field of view that was designed to detect and localize transient events. When a transient is detected, either on-board or externally, the BAT saves time-tagged event (TTE) data which provides the highest quality information of the locations of the photons on the detector plane and their energies. This data can be used to produce spectra, lightcurves, and sky images of a transient event. While these data products are produced by the Swift Data Center and can be produced by current software, they are often preset to certain time and energy intervals which has limited their use in the current time domain and multi-messenger environment. Here, we introduce a new capability for the *BatAnalysis* python package to download and process TTE data under an open-source pythonic framework that allows for easy interfacing with other python packages. The new capabilities of the *BatAnalysis* software allows for TTE data to be used by the community in a variety of advanced customized analyses of astrophysical sources which BAT may have TTE data for, such as Fast Radio Bursts (FRBs), Gamma-ray Bursts (GRBs), Low Mass X-ray Binaries (LMXB), Soft Gamma Repeaters, magnetars, and many other sources. We highlight the usefulness of the *BatAnalysis* package in analyzing TTE data produced by an on-board GRB trigger, a FRB external trigger, a sub-threshold detection of the LMXB EXO 0748-676, and an external trigger of a GRB that BAT detected during a slew.

1. INTRODUCTION

The Neil Gehrels Swift Observatory (Gehrels et al. 2004) launched on November 20th, 2004 with three telescopes onboard. These are the X-ray Telescope (XRT; (Burrows et al. 2005)), the Ultraviolet-Optical Telescope (UVOT; (Roming et al. 2005)), and the Burst Alert Telescope (BAT; Barthelmy et al. (2005)). The Swift BAT was designed to detect and localize Gamma-Ray Bursts (GRBs) and has significantly advanced our understanding of these transient events. Besides GRBs, BAT has also triggered on a number of known sources, such as soft gamma repeaters, as well as other transients, such as tidal disruption events.

The BAT uses the coded mask technique to produce small localization regions (∼ 3 arc-minutes) and accurate background estimations while maintaining a large field of view ($\sim 60^\circ \times 120^\circ$). When a transient triggers BAT on-board, the spacecraft autonomously slews such that the narrow field telescopes can observe the region of the sky where the transient was localized to. Additionally, the associated BAT data is downlinked to the

Corresponding author: Tyler Parsotan
tyler.parsotan@nasa.gov

ground. This data is the time-tagged event (TTE) data which records all measured photons around the time of the triggering event, providing the location on the detector plane where each photon was detected and each photon's energy. Besides on board triggers, TTE data can be obtained using the GUANO system which commands the spacecraft to save TTE data surrounding the time of an external trigger (Tohuvavohu et al. 2020). Furthermore, TTE data is saved for "failed" triggers or events that didn't meet the triggering threshold for BAT to autonomously trigger and slew to the localized position of the transient.

The BAT TTE data has been used in the literature for many different sources and analyses. The TTE data for BAT triggered GRBs were analyzed to produce the first, second, and third Swift Gamma-ray Burst (GRB) catalogs (Sakamoto et al. 2008, 2011; Lien et al. 2016) with pre-processed GRB lightcurves and spectra for the community to use. BAT TTE data is additionally used alongside XRT and UVOT data to provide a comprehensive view of GRBs from their prompt to afterglow emission (Evans et al. 2010). In depth analyses of the TTE data for GRBs has shown the presence of a lag–luminosity correlation for GRBs (Ukwatta et al. 2010; Parsotan & Ito 2022) which can be used to constrain the physics of GRB jets. BAT TTE data has additionally been used to measure quasi-periodic oscillations from the magnetar SGR 1806–20 during a giant outburst (Palmer et al. 2005) and, more recently, from GRB 211211A (Chirenti et al. 2024). On longer timescales, BAT TTE data was used alongside NICER data to study the outburst spectra from the magnetar Swift J1555.2-5402 (Enoto et al. 2021) over a course of a month while many years of TTE data was accumulated to analyze the fluctuations of the Crab and show that there is an anti-correlation between its flux and spectral photon index (Augustine & Chang 2024). The expanded capability of obtaining BAT TTE data for external triggers using the GUANO system has also allowed that data to be used to localize transient events using the NITRATES low-latency search, advancing the science return of many complimentary observatories (DeLaunay & Tohuvavohu 2022). In the search for electromagnetic emission from gravitational wave sources, BAT TTE data has played a critical role in placing constraints on the expected rate of coincident gravitational wave-electromagnetic radiation detections from binary Black Hole mergers and Neutron Star-Black Hole mergers (Ronchini et al. 2024; Raman et al. 2024).

BAT TTE data plays a critical role in the current time domain and multi-messenger landscape, however its use has been limited by the tools that have been made to analyze this data type. The method to analyze BAT TTE data in the context of GRBs had been made public through the HEASoft `batgrbproducts` pipeline script, however this script does not allow for full flexibility in the different timing and energy parameters that a user may want to customize. Furthermore, this pipeline cannot handle the analysis of TTE data that is collected due to "failed" triggers or by the GUANO ground system. To make the full set of TTE data more accessible and modernize the analysis of this data in the context of the greater python environment, we have modified the *BatAnalysis* python package (Parsotan et al. 2024, 2023) to be able to handle the analysis of TTE data alongside BAT survey data.

The structure of this paper is as follows. Section 2 outlines the general steps taken to analyze BAT TTE data. Section 3 discusses the *BatAnalysis* python package and how it can be used to analyze TTE data. Section 4 presents the results of our *BatAnalysis* code as compared to prior analyses and shows new analyses that are possible with the software. Finally, in Section 5, we conclude with future improvements of the *BatAnalysis* code.

2. BAT TTE DATA

Here, we describe the general methodology of how BAT TTE data is collected and processed.

2.1. *Obtaining TTE Data*

The most well known method for obtaining TTE data from BAT is through an on-board detection of a transient. As BAT is detecting photons, the flight software constructs count rate lightcurves on different timescales and energy ranges. If the software detects an increase in rates it constructs an image of the sky and compares the location of the potential source of interest to an on-board catalog of known sources[1]. If there is a new source in the image, then this triggers the observatory to slew to the calculated position of the new source in the constructed image. As BAT is surveying the sky it is already producing these images of the sky with different integration times and scaling them, which are referred to as scaled maps. An alternative way for BAT to trigger is to detect a new source within these scaled maps greater than some signal-to-noise ratio (SNR) threshold. Thus, BAT can trigger based on a rate trigger, where an increase in count rate compared to the background meets

[1] This on-board catalog can be added to as new sources are detected. This prevents known sources from constantly re-triggering BAT unless the SNR of their detection exceeds some predetermined threshold.

a SNR threshold, or the detection of a new source in a scaled map of the sky. In either case, TTE data starting from before the trigger time to after the trigger time will be saved and downlinked; usually this is 10-15 minutes worth of TTE data. Furthermore, it is important to highlight that BAT can only trigger on a transient when the spacecraft is not slewing.

A capability that is not as well known is BAT's ability to save TTE data around some "failed" or subthreshold trigger. This is an event that the onboard software detects as meeting the on-board rate trigger threshold but does not surpass the on-board imaging SNR threshold. As a result, no actual trigger will occur where the observatory will slew. However, in this situation the on-board software will save a small amount of TTE data around the time of the detected rate increase. This amount of data is correspondent to $\sim$ 10 seconds. This situation can crop up if a transient is simply not bright enough in the image domain or if a source listed in the on-board catalog does not produce a transient event that exceeds it's preset SNR threshold to trigger BAT.

Finally, Tohuvavohu et al. (2020) developed the GUANO ground system, which permits BAT TTE data to be captured for external triggers such as those from the Fermi Gamma-ray Burst Monitor (Meegan et al. 2009). The GUANO system ingests the time of the external trigger and sends a command to BAT telling it to save some portion of TTE data surrounding the trigger time before it gets overwritten on-board[2]. The commanding is successful $>90\%$ of the time which means that either 200 or 90 seconds of TTE is later downlinked to the ground.

2.2. *Processing TTE Data*

The processing of the BAT TTE data is implemented in the `batgrbproduct` HEASoft script. This script processes TTE data that was collected by an on-board trigger using predetermined energy and temporal ranges for spectra and lightcurves. Here, we briefly describe the steps followed by this script, which is also shown in a high level flow chart in Figure 1 [3]. While we mention of some aspects of coded-mask instrumentation and data analysis techniques, a more in-depth discussion of the various techniques can be found in Braga (2019) and Goldwurm & Gros (2024) and the references therein.

TTE data is a list of events that have been detected on the BAT detector plane, where an event refers to something that generates a count on the detector plane such as a photon or a cosmic ray. This list contains the event's energy and the location on the detector plane where it was measured. The location is especially critical as it allows for the coded mask weighting to be calculated and images to be constructed.

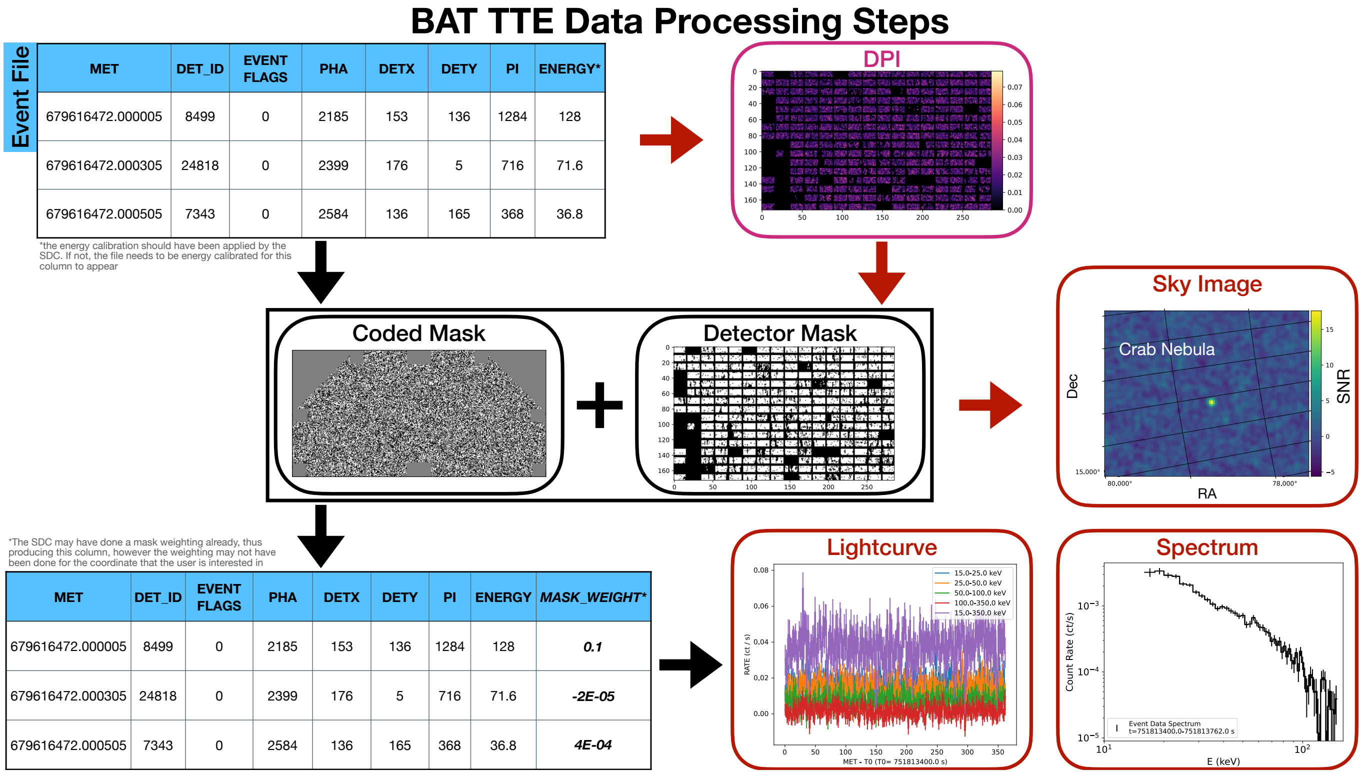
The HEASoft script `bateconvert` ensures that the TTE data has the appropriate energy calibration applied to it. It uses BAT housekeeping data to apply the appropriate gain and offset calibrations. This task is typically done by default by the Swift Data Center (SDC) when it is processing the triggered TTE data from the BAT.

An additional complication is the fact that some portion of BAT's 32,768 detectors may be disabled or actively noisy. These detectors can be turned off by the instrument team to address operational issues or they can be turned off by the flight software if it identifies a given detector as being noisy. This data is reflected in the global quality mask, which is updated in the HEASARC's Calibration Database (CALDB), and the enable/disable map, which is automatically included in the observation ID dataset that is downloaded. Sometimes, detectors become noisy during a trigger. These detectors can be identified by creating a detector plane image (DPI) from the event data and using the HEASoft script `bathotpix` to identify which detectors are noisy during the trigger. All of these different maps of disabled or noisy detectors are combined to give us the detector quality mask which is incorporated into the analysis of the BAT TTE data.

A TTE dataset from a triggered event will be processed by the SDC and, by default, the events will be weighted by the location of the event that triggered BAT. This location is the position identified by the onboard flight software which gets downlinked via the Tracking and Data Relay Satellite System (TDRSS). The mask-tagging algorithm identifies which rays of light from the location are able to propagate to the detector plane without being attenuated by the coded mask. Detectors that receive no photons from the position on the sky get a weight, $w=-1$, while detectors that are fully illuminated get assigned a weight, $w=1$. Partially illuminated detectors are assigned weights between -1 and 1 based on the proportion of the detector that is illuminated. This weighting process transforms the Poisson statistics associated with counting photons on the detector plane to Gaussian statistics. Furthermore, it means that the resulting lightcurve and spectra produced from the weighted event data correspond to

[2] The status of these commands and the GUANO system can be found at: https://www.swift.psu.edu/guano/. This webpage also lists the different types of external triggers that motivated the GUANO data dump

[3] Additional information can also be found at: https://swift.gsfc.nasa.gov/analysis/bat_swguide_v6_3.pdf and https://swift.gsfc.nasa.gov/analysis/BAT_GSW_Manual_v2.pdf



MET	DET_ID	EVENT FLAGS	PHA	DETX	DETY	PI	ENERGY*
679616472.000005	8499	0	2185	153	136	1284	128
679616472.000305	24818	0	2399	176	5	716	71.6
679616472.000505	7343	0	2584	136	165	368	36.8

*the energy calibration should have been applied by the SDC. If not, the file needs to be energy calibrated for this column to appear

*The SDC may have done a mask weighting already, thus producing this column, however the weighting may not have been done for the coordinate that the user is interested in

MET	DET_ID	EVENT FLAGS	PHA	DETX	DETY	PI	ENERGY	*MASK_WEIGHT**
679616472.000005	8499	0	2185	153	136	1284	128	***0.1***
679616472.000305	24818	0	2399	176	5	716	71.6	***-2E-05***
679616472.000505	7343	0	2584	136	165	368	36.8	***4E-04***

Figure 1. A flowchart outlining the different steps taken to obtain various high level data products from BAT TTE data corespondent to a Crab calibration observation. Starting with an event file that has been energy calibrated (which is typically done at the Swift Data Center), a user can create a DPI. The DPI is cross correlated with the coded mask to produce the sky image. This path is represented by the red arrows in the flowchart. To produce the mask-weighted lightcurve and spectra for the source of interest, the event file has to be modified by appending the file with a new column that contains the calculated mask weights. Then, the lightcurves and spectra can be calculated by binning the event data into their respective time and energy bins. These steps are represented by the black arrows on the flowchart. As a note, the Swift Data Center (SDC) typically does the energy calibration for the event data but if this is not done, the users will need to apply the energy calibration themselves. Similarly, the SDC may already calculate the mask weights for a position on the sky, which will be reflected in the event file. If the location on the sky that was used in calculating the mask weights is not the coordinates of the source that the user is interested in, then they will need to redo the mask weighting themselves.

just the source of interest. In other words, the weighting effectively removes the background due to diffuse X-ray emission as well as other sources in the field of view[4]. Changing the location on the sky for which mask weighting is performed will eventually produce a new background subtracted lightcurve and spectra for that new position in the sky. The HEASoft `batmaskwtevt` script does the mask weighting for a TTE dataset, taking a number of corrections into account including the number of active detectors and the geometrical corrections needed for a given source position with respect to the instrument boresight. This is shown in Figure 1 by following the black arrows on the left of the figure connecting the energy calibrated event file to the mask weighted event file. Not only does this script modify the event file, but it outputs the auxiliary ray tracing file which traces the position of the coordinate of interest on the detector plane as the spacecraft slews. This information is necessary for analyzing spectra, especially those that overlap with a slew.

Constructing the lightcurves and spectra for the single position in the sky is simply a sum of all the events multiped by their weights [5]. The HEASoft `batbinevt` script does this operation for both lightcurves and spectra, and is represented in Figure 1 by the black arrow that connects the modified event file to the mask-weighted lightcurve and spectra. The spectra require additional

[4] The removal of background sources with this method is not complete as photons from other sources in the FOV can still be detected at a given location on the detector plane where the associated weight is non-zero. Thus, there is technically still some contamination from other sources with the mask-weighting method.

[5] There is an implicit unit of 'per detector' that gets applied to the lightcurves and spectra units when the weights are applied such that the units of counts/s really mean counts/s/detector.

information related to the location of the sky coordinate with respect to the detector boresight and the systematic error on the spectra. The geometric information of the sky coordinate position on the detector plane is taken from the auxiliary ray-tracing file that was previously created and inserted into the spectrum file using the `batupdatephakw` HEASoft script. The HEASoft script `batphasyserr` obtains the BAT systematic error for each energy channel from CALDB and appends it to the spectral file. Finally, to construct a detector response matrix (DRM) we use the `batdrmgen` HEASoft script which should be used to construct a response for each spectrum where the source of interest is at a different location within the BAT field of view, for example each spectrum that is constructed during a slew.

Sky images can also be constructed from the BAT TTE data. This is done by creating DPIs for the time interval of interest and using the `batfftimage` HEASoft script to deconvolve the DPI with the BAT coded aperture mask taking a number of coded mask and geometric corrections into account. The `batfftimage` script can also be used to construct the partial coding map of the sky, which represents the percentage of the BAT detector plane that is exposed to a given point on the sky. These steps are represented as the path that follows the red arrows in Figure 1 connecting the event file, the DPI, the coded mask and detector mask, and the resulting sky image.

The `batgrbproduct` script processes a given TTE dataset that was produced by an onboard trigger and produces lightcurves, spectra, detector responses, and images following the methodology outlined above. The default weighted, or background subtracted, lightcurves produced by `batgrbproduct` are binned into either a single (15-350 keV) energy channel or 4 energy channels (15-25, 25-50, 50-100, and 100-350 keV) in time bins of 64 ms or 1 s. Additional lightcurves that are generated are a 15-350 keV lightcurve in 4 ms timebins, an unweighted 15-350 keV lightcurve binned into 4 ms timebins, and a single and 4 energy channel Bayesian block lightcurve constructed using the `battblocks` HEASoft script. The output of `battblocks` also includes estimates of the trigger's time to accumulate 90%, 50% and 100% of the photons from the triggering source (T_{90}, T_{50}, and T_{100} respectively), as well as the peak 1 s time interval, and the background time intervals. The spectra that `batgrbproduct` creates by default have 80 energy channels, between 15-150 keV[6], and they are constructed for the various time intervals identified with `battblocks`. Additional spectra are constructed for the pre-slew, post-slew, and slew time periods. The detector responses that are created correspond to the pre-slew and post-slew time periods. Users can only use these responses when fitting models to spectra that were constructed for these pre- and post-slew time periods. `batgrbproduct` also produces DPIs and sky images for the same pre- and post-slew time intervals in both single and 4 energy channels. DPIs and images are constructed for the time interval prior to any detectable emission from the source, which is denoted at the “pre-burst” time interval. Finally, partial coding maps are constructed for the pre- and post-slew time intervals.

The `batgrbproduct` script comprehensively takes a number of calibrations and coded-mask corrections into account to produce the high-level lightcurves, spectra, detector responses, DPIs, and sky images which users can use directly. However, the preset time and energy binning make the output incompatible with joint spectral analysis studies or timing studies of the lightcurves. Furthermore, this pipeline does not handle processing subthreshold trigger TTE data or GUANO TTE data.

3. THE *BatAnalysis* CODE

In this section we outline the *BatAnalysis* (Parsotan et al. 2024) python package's TTE data analysis capability[7]. The *BatAnalysis* package allows a user to:

1. Download BAT triggered, subthreshold, and GUANO TTE data
2. Process the BAT TTE dataset
3. Construct custom lightcurves, spectra, and images
4. Search images for flaring sources
5. Conduct spectral fitting

The *BatAnalysis* package is a wrapper for HEASoftpy which allows for the identical processing of BAT TTE data as what is outlined in Section 2.2. Thus, the files

[6] Technically, BAT has a non-coded response up to 500 keV but the 15-150 keV energy range is recommended for conducting spectral fits with BAT event data since the coded mask becomes transparent above $\sim$ 150 keV. Since the spectra are constructed by mask-weighting the TTE data this technique only works when the mask is opaque to the detected radiation, in the 15-150 keV band. Additionally, the detector responses are not calibrated above 150 keV (DeLaunay & Tohuvavohu 2022) further complicating the spectral fitting above this energy.

[7] This package is open source and is available on github at: https://github.com/parsotat/BatAnalysis (Parsotan et al. 2024)

produced by the processing of the TTE data are also identical to what is produced by the HEASoft scripts. The *BatAnalysis* package allows for the high level BAT TTE data products to be integrated into the greater astronomical data analysis python environment. In this section we elaborate on each of the prior highlighted capabilities of the software.

The documentation included in the *BatAnalysis* github repository goes into the details of how to utilize the software to accomplish the tasks outlined in this section. Furthermore, the codes that were used to produce the results in Section 4 provide examples of how to use the python package to process the different TTE datasets.

3.1. *Downloading BAT TTE datasets*

Prior methods of querying and downloading BAT datasets required using the High Energy Astrophysics Science Archive Research Center (HEASARC) for observations correspondent to a trigger that a user is interested in and then downloading the datasets through their web interface. Downloading GUANO or subthreshold trigger is less straightforward as these TTE data types are archived using different schemes[8] and can be difficult to query based on time or position criteria. The *BatAnalysis* python package interfaces with the swifttools (Kennea et al. 2024) python package, to allow for querying and downloading observation IDs based on different criteria. Users are able to search and download "normal" triggered TTE data, GUANO downlinked TTE data, and subthreshold TTE data. In particular, the *BatAnalysis* software takes into account the need for temporally associated energy calibration that typically has to be applied to subthreshold TTE data.

3.2. *Processing BAT TTE observations*

The steps for processing TTE data are identical to the ones outlined in Section 2.2. As a result, it calls many of the HEASoftpy scripts that are needed to process the data including performing energy calibration, determining bad detectors, applying mask weighting, and constructing DPIs, lightcurves, spectra, and images. Since these same HEASoft scripts are called in the background, the output files are identical to those that are produced using the steps previously outlined. Users can additionally pass a dictionary that has key/value pairs that correspond to the HEASoft parameters and the values that the user would like to set for those HEASoft script parameters. Overall, much of the documentation that is relevant for the various HEASoft scripts is still relevant for the *BatAnalysis* software.

There are major advantages with processing data using *BatAnalysis*. The pythonic interface allows users to dynamically rebin DPIs, lightcurves, and spectra in energy and time while offering plotting interfaces that utilize matplotlib for quickly viewing these different products and eventually producing publication ready figures. Furthermore, the images that are generated by *BatAnalysis* can easily be projected onto a healpix map making them more compatible with modern astrophysical analyses and mosaiced to enhance any detected sources in images.

3.3. *Spectral Fitting*

Once a user has created a PHA file with *BatAnalysis*, the appropriate corrections are automatically applied and an associated DRM gets automatically created. The PHA can be fitted within the *BatAnalysis* software through the use of pyXspec, similar to the capabilities that are available for spectra obtained for survey data. With pyXspec, *BatAnalysis* will fit a model to the spectrum utilizing Gaussian statistics. The default model is a `cflux*powerlaw` model that fits the spectra from 15-150 keV and users can pass in their own spectral models based on what they may expect. Additionally, the user can use the PHA and DRM files created by *BatAnalysis* in their own spectral fitting tools but special care may be needed to ensure that the appropriate systematic errors are properly applied.

Using the spectral fittings, users can determine if the source is significantly detected following the same method used for survey data (Parsotan et al. 2023). A flux upper limit can be determined by scaling the background variation by the significance level that the user is interested in and then fitting the resulting spectrum to obtain the flux needed to make such a measurement (Parsotan et al. 2023; Laha et al. 2022).

As is highlighted by Lien et al. (2016), the time period that a user would like to construct and fit spectra may overlap with a spacecraft slew. In this case, Lien et al. (2016) constructed an averaged DRM over the time period where Swift is slewing. This averaged DRM can be constructed by taking the weighted average of many DRMs that were constructed on smaller time intervals. The weighting can be equal to the fraction of photon counts in the smaller time intervals where the individual DRMs are constructed. Using the *BatAnalysis*, the construction of such a time averaged DRM is possible and its use in spectral fittings is convenient.

[8] This archiving scheme is documented in the "Overview of the Swift Data Archive" document available at https://swift.gsfc.nasa.gov/archive/

3.4. *Mosaiced Images*

It is possible to interpolate the images produced using BAT TTE data onto the sky and add them together, identical to what is done with BAT survey data. This technique can be utilized in a similar method to the dithering technique used in narrow field instruments, where images of a source which is moving through the FOV are stacked leading to an increased SNR of the source of interest. In the case of GUANO downlinked TTE data, this capability also allows for searches of transient sources while BAT is slewing which is particularly important as Swift spends a larger portion of time slewing from source to source as a target of opportunity focused facility. When Swift is slewing the triggering is turned off for BAT which has an adverse effect on the number of on-board triggered GRBs since Swift has become more driven by target of opportunity observations.

Mosaicing BAT TTE images is straightforward with the *BatAnalysis* package. Users first construct sky images as usual and then these sky images can be added together. Behind the scenes, the *BatAnalysis* is linearly interpolating each sky image onto a healpix map of a given resolution and then adding each healpix map together. The different sky images are added identical to the formalism used for survey data except there is no energy dependent off-axis correction which takes the width of the coded mask into account (Parsotan et al. 2023). The total flat exposure for each pixel in the mosaic healpix map, i, is calculated as:

$$E_i = \sum_k e_{i,k} \tag{1}$$

where E_i is the mosaic image's flat exposure in each pixel and $e_{i,k}$ is the flat exposure map of the k^{th} sky image that will be included in the mosaic once it has been interpolated on the healpix map. To calculate the partial coding vignetting mask we multiply the flat exposure by the partial coding once each sky image has been interpolated onto the healpix map and add the values in each pixel. This is given as

$$P_i = \sum_k e_{i,k} p_{i,k} \tag{2}$$

where P_i is the mosaic partial coding exposure map and $p_{i,k}$ is the partial coding map for the k^{th} image that is being mosaiced once it has been interpolated onto the healpix grid.

The standard deviation of the mosaiced image for energy band l, $\sigma_{i,l}$, is calculated by adding the inverse variances and then converting back to normal standard deviation. This is given as:

$$\sigma_{i,l} = \frac{1}{\sqrt{\sum_k v_{i,k,l}^{-2}}} \tag{3}$$

where $v_{i,k,l}$ is the energy dependent standard deviation map of each individual image once it has been interpolated onto the healpix grid.

To calculate the mosaiced flux sky image for energy band l, $S_{i,l}$, we add the flux sky images weighted by their associated inverse variance map. This is given as:

$$S_{i,l} = \sigma_{i,l} \sum_k s_{i,k,l}\ v_{i,k,l}^{-2} \tag{4}$$

where $s_{i,k,l}$ is the flux sky image in each energy band that the user wishes to mosaic together interpolated on the healpix grids before being multiplied by $v_{i,k,l}$.

These SNR in each pixel of the mosaic healpix map at a given energy can then be calculated simply as

$$\text{SNR}_{i,l} = \frac{S_{i,l}}{\sigma_{i,l}} \tag{5}$$

Sources in the mosaiced images can be searched based on the SNR of each pixel where typically SNR values > 7 can be trusted. As an additional check for new sources, the healpix pixel's distance from known sources can be checked and compared to the full width half maximum size of the BAT's point spread function. This check ensures that potential high SNR pixels in the mosaic are distinct from known sources that have been previously detected by BAT. Copete (2012) outlines many of the details needed to be considered in mosaicing together images from TTE data with the goal of detecting and localizing transient sources.

3.5. *Parallelized Analysis*

There are additional capabilities offered in *BatAnalysis* to allow users to expedite their analyses through parallelized functions. These functions allow users to perform the following operations in parallel: produce spectra, fit spectra, produce images, and mosaic images. While these functions are simplified to permit straightforward analyses in a parallelized manner, users can use them as templates for their own personal analyses with differing analysis requirements.

4. APPLICATIONS OF BAT TTE DATA

In this section, we validate the software by analyzing the on-board triggered GRB 050724, a short GRB with extended emission. Additionally, we show the capabilities of the *BatAnalysis*code as applied to BAT GUANO data coincident with a FRB external trigger from FRB 180916, subthreshold trigger data that was collected for

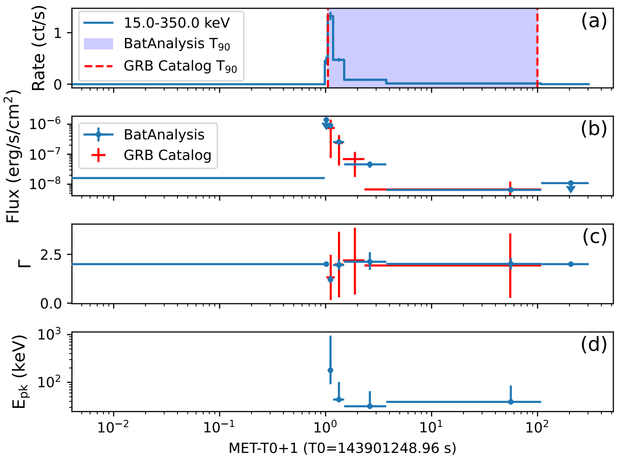


Figure 2. The mask-weighted lightcurve (with implied rate units of counts/s/detector) and fitted time-resolved spectral parameters for GRB 050724, including estimates of the spectral E_{pk}. In panel (a), we plot the *BatAnalysis* Bayesian binned lightcurve and we highlight the measured T_{90} in the blue highlighted region. We also show the T_{90} published in the GRB BAT Catalog for this event. In Panel (b) and (c) we show the *BatAnalysis* fitted spectral flux and power law photon index respectively in blue. We plot the spectral fits for the spectra provided in the GRB Catalog in red. In panel (d) we plot the estimated spectral E_{pk} (Sakamoto et al. 2009). To better highlight both the initial short pulse and the extended emission, the times are shifted by 1 second with respect to the trigger time.

the LMXB EXO 0748-676, and BAT GUANO data that was saved due to the Fermi trigger of GRB 210706A[9].

4.1. *GRB 050724 - On-board triggered TTE Data*

The first science case that we use to exemplify the capabilities of *BatAnalysis* and verify the software is with the analysis of GRB 050724. This unique GRB is a short GRB with extended emission due to the short pulse that lasts for less than 2 seconds however the measured T_{90} is significantly longer at $\sim$ 100 seconds. BAT was triggered by this event on 2005-07-24 at 12:34:09.361840 UTC and promptly slewed to the coordinates determined on-board for this burst.

In Figure 2 we show the mask weighted lightcurve and the time resolved spectral parameters as a function of time. Both the lightcurve and the spectra are offset from the trigger time by 1 second in order to better show both the short pulse and the extended emission. In Figure 2(a), we show the Bayesian binned count rate lightcurve in the 15-350 keV energy range. We highlight the T_{90} obtained by *BatAnalysis* with the blue highlighted region as well as the T_{90} published for this GRB on the BAT GRB Catalog website (Lien et al. 2016). In Figure 2(b) and Figure 2(c) we plot the flux and the power law index obtained from fitting the time resolved spectra. The blue points show the quantities obtained by fitting the spectra using their associated DRMs produced by *BatAnalysis* with a `cflux*powerlaw` model for each timebin of the Bayesian binned lightcurve. The red points are the time resolved spectra obtained from the BAT GRB Catalog for this GRB. The time binning is slightly different and the spectra that are produced are fitted with a `cflux*powerlaw` using the time averaged DRM that is available on the catalog webpage for this GRB. In Figure 2(d), we use the *BatAnalysis* `cflux*powerlaw` spectral fit parameters to estimate the spectral peak energy, E_{pk}, using the method outlined by Sakamoto et al. (2009). We find that the *BatAnalysis* software is able to recover the duration of this GRB and the time resolved spectral parameters, albeit with smaller errors in the fitted parameters and a $\sim$ 1 second difference in the start and end time of the timebins that encompass the extended emission. The increased precision of the *BatAnalysis* spectral parameters is due to the fact that *BatAnalysis* uses the DRM that is constructed for the same time interval that is used to construct the PHA file. This is an improvement over the spectral analysis presented in the GRB catalog where the time averaged DRM is used in all spectral fittings. This is only necessary for the last timebin of the extended emission phase where Swift is slewing.

We also directly verify the spectra that are obtained within the T_{90} time range in Figure 3, which encompasses a time period when Swift is slewing. Here, we plot the *BatAnalysis* calculated 80 channel spectrum plotted for the 15-150 keV energy range in black with the 1σ errors in each channel. We construct a time averaged DRM and use it to fit the *BatAnalysis* T_{90} spectrum using a `cflux*powerlaw` model and show the folded model spectrum in red. Finally, we plot the T_{90} spectrum in the GRB catalog in the same 15-150 keV energy range in blue points. We can see that the *BatAnalysis* spectrum agrees well with the spectrum published in the BAT GRB catalog, verifying the correctness of the software.

4.2. *FRB 180916 - GUANO TTE Data*

The next science case that we use to showcase the capabilities of the *BatAnalysis* python package is analyzing GUANO downlinked TTE data correspondent with an external detection of FRB 180916. The *BatAnalysis* package makes it straightforward to process this data

[9] All the code for the examples shown here can be found in the notebooks subdirectory of the github repository: https://github.com/parsotat/BatAnalysis/tree/main/notebooks.

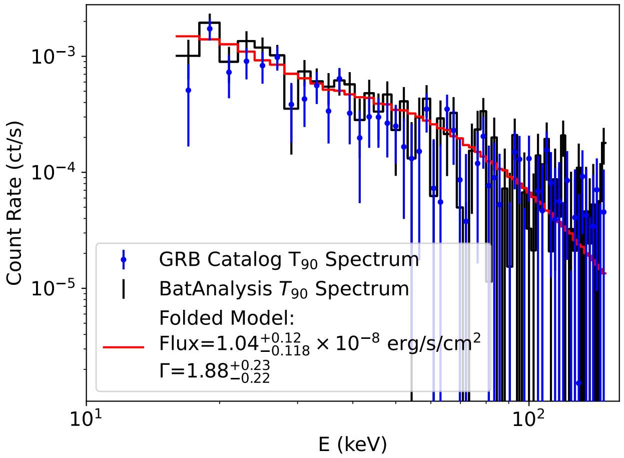


Figure 3. The *BatAnalysis* produced T_{90} spectrum, plotted in black, agrees with the T_{90} spectrum obtained in the BAT GRB catalog for this GRB, plotted in blue. Since this spectrum overlaps a time period where Swift is slewing we construct a time averaged DRM which we use in our spectral fitting. The best fit `cflux*powerlaw` model to the *BatAnalysis* spectrum is shown in red with the spectral parameters provided in the legend.

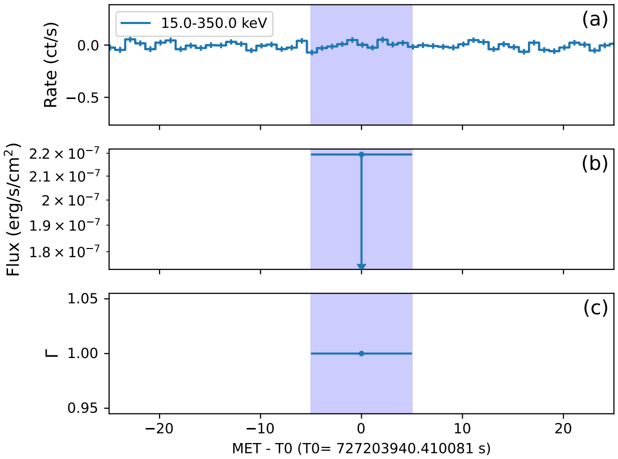


Figure 4. We plot the mask weighted lightcurve for the location of the FRB in the 15-350 keV energy range in panel (a) and also highlight, in light blue, the T0±5 s time period where we place upper limits on the FRB emission. In panel (b) we show the flux upper limit obtained for the FRB in the time period of T0±5 s. In panel (c) we plot the photon index, Γ, used to fit a `powerlaw` model to the upper limit spectrum to obtain the upper limit.

and eventually place flux upper limits on the FRB in the 15-150 keV energy range.

We select a GUANO dataset where the partial coding for the coordinates of the FRB is $\sim$ 19%. The dataset that we analyze is correspondent to the FRB trigger time on 2024-01-17 at 17:05:07.661808 UTC and the GUANO data is associated with Observation ID 01209406000. The results of the mask weighted lightcurve calculation and the flux upper limit that is obtained for the FRB is shown in Figure 4. In Figure 4(a), we show the mask weighted 15-350 keV lightcurve for the FRB, showing the 200 seconds of data that was saved by the GUANO system and the lack of a clear signal around the FRB trigger time. As as result, we use the *BatAnalysis* software to place upper limits on the FRB emission in the 15-150 keV energy range from T0-5 seconds to T0+5 seconds. We calculate the 5σ upper limit spectrum following the method outlined in section 3.3 and fit a powerlaw with a photon index of 1. Our flux upper limit of $\sim 2.2 \times 10^{-7}$ erg/s/cm^2 is shown in Figure 4(b) and the power law photon index used to obtain the upper limit is shown in Figure 4(c).

4.3. *LMXB EXO 0748-676 - Subthreshold Trigger TTE Data*

We use the subthreshold trigger that occurred with LMXB EXO 0748-676 to showcase the *BatAnalysis* package's ability to facilitate the finding and analysis of this type of TTE data.

In July 2024, there were a set of subthreshold triggers that were due to this source (Palmer & Swift/BAT Team 2024). Since this source is known and the outbursts did not meet the trigger thresholds for this source (except for one outburst which triggered BAT (D'Elia et al. 2024)) the subthreshold TTE data was saved by the onboard software. We analyze data for subthreshold trigger #1245446, which occurred on 2024-07-29 at 12:05:16 UTC to show that this source is actually flaring, as is shown in Figure 5. Here, we show the BAT's field of view in galactic coordinates on a healpix map. The plotted quantity is the SNR map in the 15-25 keV energy band. The inset plot, shows the location of EXO 0748-676 and the SNR peak associated with the LMXB with SNR $\sim$ 7.

In Figure 6(a) we show the mask weighted lightcurve of the source. We can see a slow rise in the lightcurve for the data that is analyzed. We use the time resolved spectra to place flux upper limits on the LMXB in the 15-150 keV energy range, which is shown in Figure 6(b). The photon index, Γ, used to fit the upper limit spectra with a `powerlaw` model is shown in Figure 6(c). We find that the flux of the LMXB is $\lesssim 3 \times 10^{-7}$ erg/s/cm^2 in all 1 second time bins.

4.4. *GRB 210706A - Slew Imaging Analysis*

In this section we show that the *BatAnalysis* can be used to analyze a GRB that was detected in GUANO TTE data while Swift was slewing. We consider GRB

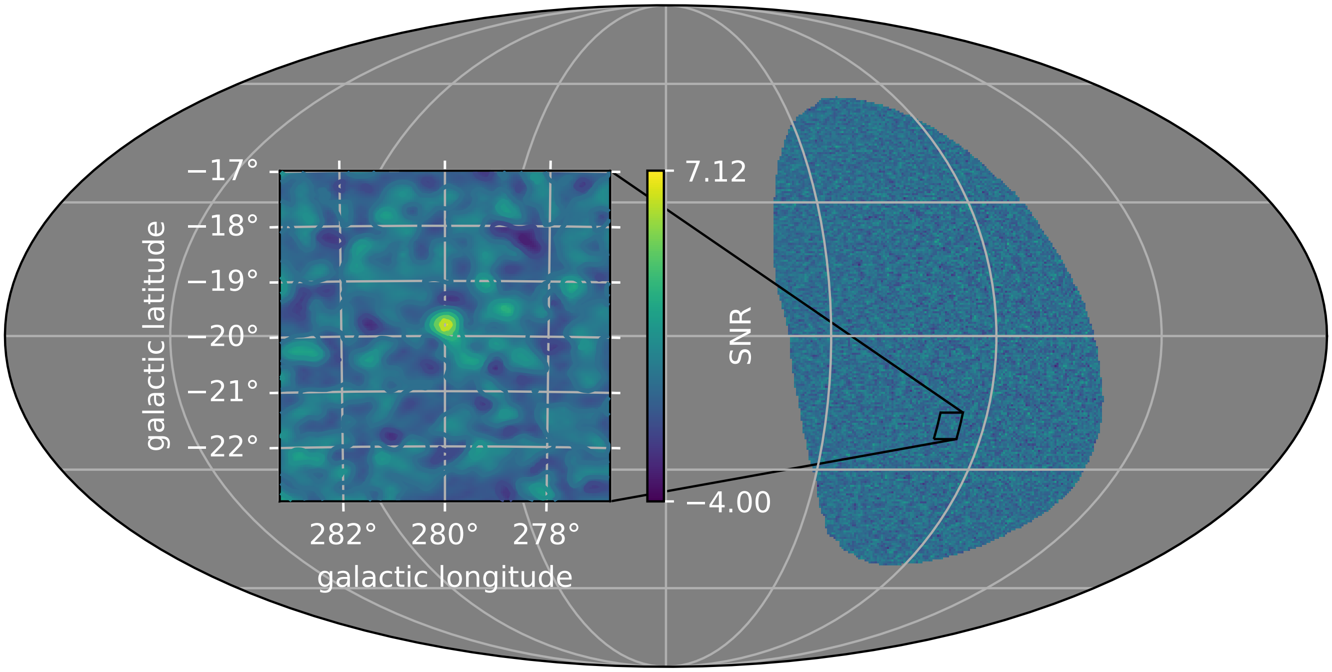


Figure 5. Using the subthreshold trigger TTE data, we show that there is a SNR excess at the location of EXO 0748-676. We plot the 15-25 keV SNR map of the sky, in galactic coordinates, as viewed by BAT at the time in a mollewide projection. We also plot a zoomed in view of the location of the LMXB, showing the SNR of the source which caused the on-board software to save the subthreshold TTE data.

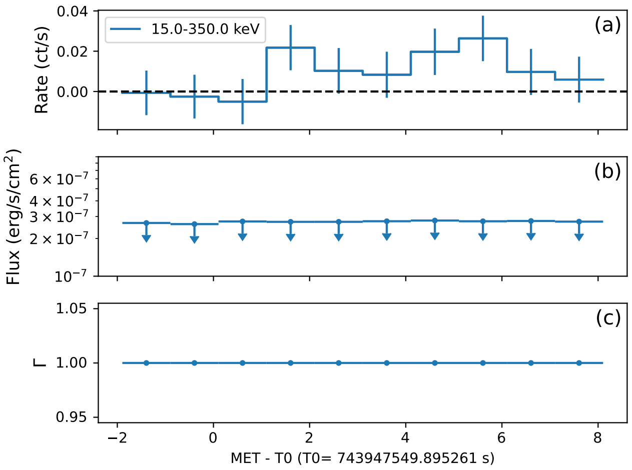


Figure 6. In panel (a) we plot the mask weighted lightcurve for EXO 0748-676 in the 15-350 keV energy range. In panel (b) we show the time resolved flux upper limits obtained for the LMXB in the time period of the subthreshold TTE data. In panel (c) we plot the photon index, Γ, used to fit `powerlaw` models to the upper limit spectra to obtain the upper limits.

210706A, detected by Fermi on 2021-07-06 at 08:17:49.9 UTC. GUANO successfully downlinked 200 s of TTE data around the GRB trigger time. From an inspection of the attitude file, we can verify that the spacecraft was slewing at T0. To localize the GRB with the TTE data, we create a mosaiced image in the energy range of 15-350 keV for the time interval spanning Δt = [T0,T0+10s], where the exposure of each image that gets mosaiced is 0.2 s following the method outlined by Copete (2012). We find a source with SNR = 39.6 at RA = 311.96 deg, Dec = 13.29 deg, compatible with the position reported in the GUANO GCN (Tohuvavohu et al. 2021a) and in the XRT refined analysis GCN (Tohuvavohu et al. 2021b). After applying the mask weighting using these coordinates, we can extract the lightcurve and spectrum for this GRB. The background-subtracted, Bayesian binned lightcurve and temporal evolution of the partial coding at the GRB position are shown in Figure 7. Using the same Bayesian block timebins, we computed the spectral evolution, using again a power law spectrum for each time bin, and the results are shown in panel (b) and (c) of Figure 7. Here, the DRM constructed for each time interval was used in the fitting of each time resolved spectrum. Using the Bayesian blocks algorithm, we obtain a T_{90} = 8.5 s, measured in the time interval t_{GRB} = [-0.42, 8.09] s. The time averaged spectrum in the time interval t_{GRB} has been fitted with a power law, obtaining a flux $(2.44 \pm 0.08) \times 10^{-7}$ erg cm^{-2} s^{-1} and a photon index 1.59 ± 0.06 in the 15-150 keV energy range. The fit was conducted using a time averaged DRM. The T_{90}

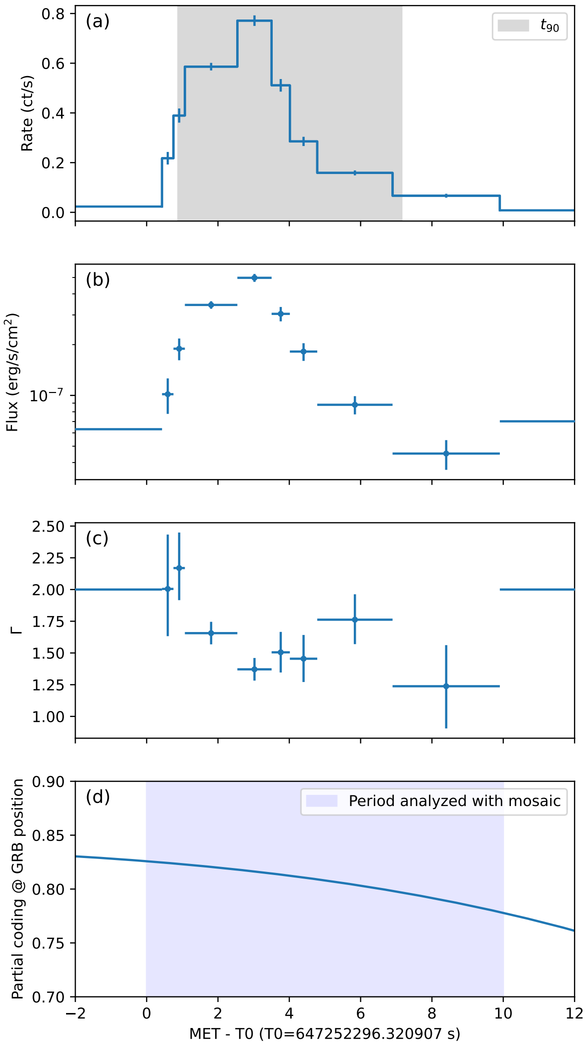


Figure 7. Results from the analysis of GRB 210706A which occurred while Swift was slewing. Panel (a): lightcurve rebinned using Bayesian blocks. The shaded gray area identifies the starting and ending time of the T_{90} interval. Panel (b) and (c): flux and photon index evolution, from time-resolved spectral analyses using a `cflux*powerlaw` spectrum. Panel (d): temporal evolution of the partial coding evaluated at the position of the GRB showing the slew that Swift was performing at the time.

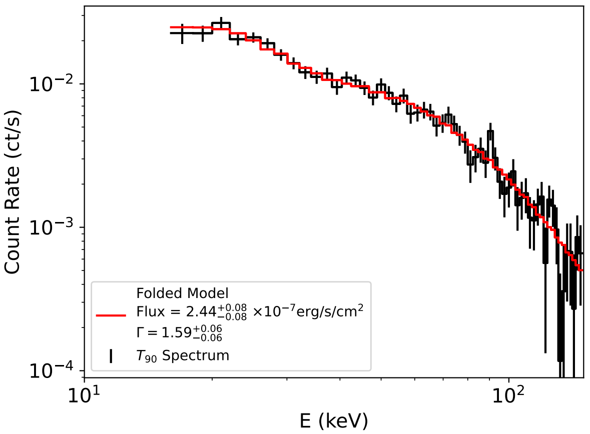


Figure 8. The time-integrated T_{90} spectrum and the fitted `cflux*powerlaw` model of GRB 210706A. The spectral fit was conducted with a time averaged DRM.

spectrum and the folded best fit model are shown in Figure 8.

5. CONCLUSIONS

We have introduced the updated *BatAnalysis* python package, which now includes the capability to analyze BAT TTE data that has been collected in multiple ways. Previously, the HEASoft tools allowed users to only analyze TTE data that was collected due to an on-board trigger. Now, the *BatAnalysis* software, allows users to process and analyze TTE data from on-board triggers, GUANO commanded TTE data dumps, and subthreshold triggers. The python package allows for more intuitive analysis of the data, and it conveniently integrates into the modern astrophysics python environment enhancing the use of BAT data in multi-wavelength and multi-disciplinary studies. We showcase the many features of the *BatAnalysis* software and verify its correctness by analyzing TTE data for GRB 050724, a short GRB with extended emission that triggered BAT. Furthermore, we show how the software can be used to analyze GUANO downlinked data, in the case of an FRB trigger for FRB 180916, a subthreshold TTE dataset obtained for an outburst of EXO 0748-676, and to localize and analyze the spectro-temporal properties of GRB 210706A, which occurred during a slew.

5.1. *Spectral Analysis Caveats*

One limitation of this software is that the spectral fitting convenience function included with *BatAnalysis* uses simple methods to fit spectra, which can sometimes give erroneous results. These results can present as unconstrained spectral parameters with large errors associated with them. Many times, adjusting various pyXspec fitting parameters such as the number iterations that the code uses to converge on a solution or the permissible ranges on spectral model parameters can help with the fitting solution. We also would like to reinforce the need to utilize gaussian statistics when conducting spectral fitting with the mask-weighted BAT spectra, which is in contrast to other gamma-ray telescopes, and the need to ensure that the spectral models used are not overfitting the data. Additionally, it is prudent to ensure that the PHA and DRM files that are used in the fits are for as identical time periods as is possible to ensure that the behavior of the instrument is captured as accurately as possible in the DRM. The additional complication of constructing a time-averaged DRM for periods where Swift may be slewing is well worth the effort as the response of BAT can change over its large field of view.

We also highlight that users can construct and use their own spectral fitting tools and spectral models within the greater python environment. This can take the form of models constructed with the astromodels package (Vianello et al. 2021) or fitting algorithms that utilize scipy minimization functions (Virtanen et al. 2020), more rigorous monte carlo markov chain (eg. using the emcee package (Foreman-Mackey et al. 2013)) or even nested sampling methods (eg using the MultiNest package (Feroz et al. 2019)), just to highlight a few.

This work adds an important data analysis capability to the *BatAnalysis* software, unlocking the wealth of BAT TTE data for general use. The *BatAnalysis* software is now a comprehensive open-source pipeline for analyzing the most important scientific data products produced by BAT. The community can contribute to the open-source project by creating a fork on github and opening a pull request to incorporate their improvements into the official version of the Python package. Furthermore, the community can post any issues that they encounter to the github repository for expedited resolution of the problems. This package may become an official part of HEASoft, in the future, making it an official Swift BAT data analysis pipeline.

ACKNOWLEDGMENTS

The material is based upon work supported by NASA under award number 80GSFC21M0002.

This material is based upon work supported by the National Aeronautics and Space Administration under Agreement No.80NSSC23K0552 issued through the Office of Science. In accordance with Federal law, NMC is prohibited from discriminating on the basis of race, color, national origin, sex, age, or disability.

This research has made use of data and/or software provided by the High Energy Astrophysics Science Archive Research Center (HEASARC), which is a service of the Astrophysics Science Division at NASA/GSFC. We thank Bryan Irby and Abdu Zoghbi for help with HEASoftpy, Israel Martinez Castellanos for his development of Histpy, and Jamie Kennea for his assistance with swifttools.

Software: Astropy (Astropy Collaboration et al. 2013, 2018, 2022), Astropy/reproject (Robitaille et al. 2024), Astroquery (Ginsburg et al. 2019), BatAnalysis (Parsotan et al. 2024) NumPy (Harris et al. 2020), Matplotlib (Hunter 2007), Scipy (Virtanen et al. 2020), HEASoft (Nasa High Energy Astrophysics Science Archive Research Center (Heasarc) 2014), swiftbat_python, Xspec (Arnaud 1996), swifttools (Kennea et al. 2024), Histpy (Martinez-Castellanos 2024)

REFERENCES

Arnaud, K. A. 1996, in Astronomical Society of the Pacific Conference Series, Vol. 101, Astronomical Data Analysis Software and Systems V, ed. G. H. Jacoby & J. Barnes, 17 5.1

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33, doi: 10.1051/0004-6361/201322068 5.1

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, aj, 156, 123, doi: 10.3847/1538-3881/aabc4f 5.1

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., et al. 2022, apj, 935, 167, doi: 10.3847/1538-4357/ac7c74 5.1

Augustine, K. A., & Chang, H.-K. 2024, arXiv preprint arXiv:2408.00406 1

Barthelmy, S. D., Barbier, L. M., Cummings, J. R., et al. 2005, Space Science Reviews, 120, 143 1

Braga, J. 2019, Publications of the Astronomical Society of the Pacific, 132, 012001 2.2

Burrows, D. N., Hill, J., Nousek, J., et al. 2005, Space science reviews, 120, 165 1

Chirenti, C., Dichiara, S., Lien, A., & Miller, M. C. 2024, The Astrophysical Journal, 967, 26 1

Copete, A. J. 2012, PhD thesis, Harvard University, Massachusetts 3.4, 4.4

DeLaunay, J., & Tohuvavohu, A. 2022, The Astrophysical Journal, 941, 169 1, 6

D'Elia, V., Kennea, J. A., Page, K. L., Parsotan, T. M., & Neil Gehrels Swift Observatory Team. 2024, GRB Coordinates Network, 36653, 1 4.3

Enoto, T., Ng, M., Hu, C.-P., et al. 2021, The Astrophysical Journal Letters, 920, L4 1

Evans, P., Willingale, R., Osborne, J., et al. 2010, Astronomy & Astrophysics, 519, A102 1

Feroz, F., Hobson, M. P., Cameron, E., & Pettitt, A. N. 2019, The Open Journal of Astrophysics, 2 5.1

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306, doi: 10.1086/670067 5.1

Gehrels, N., Chincarini, G., Giommi, P. e., et al. 2004, The Astrophysical Journal, 611, 1005 1

Ginsburg, A., Sipőcz, B. M., Brasseur, C. E., et al. 2019, aj, 157, 98, doi: 10.3847/1538-3881/aafc33 5.1

Goldwurm, A., & Gros, A. 2024, in Handbook of X-ray and Gamma-ray Astrophysics (Springer), 1613–1669 2.2

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357, doi: 10.1038/s41586-020-2649-2 5.1

Hunter, J. D. 2007, Computing in Science & Engineering, 9, 90, doi: 10.1109/MCSE.2007.55 5.1

Kennea, J., Evans, P., & Maddi, V. 2024, Swift-SOT/swifttools: swifttools-3.0.22: Add ‘srcrad‘ parameter to ‘xrt_prods‘, swifttools-3.0.22, Zenodo, doi: 10.5281/zenodo.10610946 3.1, 5.1

Laha, S., Wadiasingh, Z., Parsotan, T., et al. 2022, ApJ, 929, 173, doi: 10.3847/1538-4357/ac5f3c 3.3

Lien, A., Sakamoto, T., Barthelmy, S. D., et al. 2016, The Astrophysical Journal, 829, 7 1, 3.3, 4.1

Martinez-Castellanos, I. 2024, histpy, 1.1.1, Zenodo, doi: 10.5281/zenodo.14262814 5.1

Meegan, C., Lichti, G., Bhat, P., et al. 2009, The Astrophysical Journal, 702, 791 2.1

Nasa High Energy Astrophysics Science Archive Research Center (Heasarc). 2014, HEAsoft: Unified Release of FTOOLS and XANADU, Astrophysics Source Code Library, record ascl:1408.004. http://ascl.net/1408.004 5.1

Palmer, D., & Swift/BAT Team. 2024, The Astronomer's Telegram, 16790, 1 4.3

Palmer, D. M., Barthelmy, S., Gehrels, N., et al. 2005, Nature, 434, 1107 1

Parsotan, T., & Ito, H. 2022, Universe, 8, 310 1

Parsotan, T., Laha, S., Palmer, D. M., et al. 2023, The Astrophysical Journal, 953, 155 1, 3.3, 3.4

Parsotan, T., Palmer, D., Ronchini, S., et al. 2024, parsotat/BatAnalysis: BatAnalysis, v2.0.0, Zenodo, doi: 10.5281/zenodo.7916508 1, 3, 7, 5.1

Raman, G., Ronchini, S., Delaunay, J., et al. 2024, arXiv preprint arXiv:2407.12867 1

Robitaille, T., Ginsburg, A., Mumford, S., et al. 2024, astropy/reproject: v0.13.1, v0.13.1, Zenodo, doi: 10.5281/zenodo.10931886 5.1

Roming, P. W., Kennedy, T. E., Mason, K. O., et al. 2005, Space Science Reviews, 120, 95 1

Ronchini, S., Bala, S., Wood, J., et al. 2024, ApJ, 970, L20, doi: 10.3847/2041-8213/ad5d74 1

Sakamoto, T., Barthelmy, S., Barbier, L., et al. 2008, The Astrophysical Journal Supplement Series, 175, 179 1

Sakamoto, T., Sato, G., Barbier, L., et al. 2009, The Astrophysical Journal, 693, 922 2, 4.1

Sakamoto, T., Barthelmy, S., Baumgartner, W., et al. 2011, The Astrophysical Journal Supplement Series, 195, 2 1

Tohuvavohu, A., DeLaunay, J., Kennea, J. A., & Raman, G. 2021a, GRB Coordinates Network, 30393, 1 4.4

Tohuvavohu, A., Kennea, J. A., DeLaunay, J., et al. 2020, The Astrophysical Journal, 900, 35 1, 2.1

Tohuvavohu, A., Sbarufatti, B., Burrows, D. N., et al. 2021b, GRB Coordinates Network, 30399, 1 4.4

Ukwatta, T., Stamatikos, M., Dhuga, K., et al. 2010, The Astrophysical Journal, 711, 1073 1

Vianello, G., Burgess, J., Fleischhack, H., Di Lalla, N., & Omodei, N. 2021, astromodels, 2.2.2, Zenodo, doi: 10.5281/zenodo.5646925 5.1

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature Methods, 17, 261, doi: 10.1038/s41592-019-0686-2 5.1, 5.1