Krittika Summer Projects 2023

GRB Hunters
Faintest of the Brightest

Shreyas N B

KRITTIKA SUMMER PROJECTS 2023

GRB Hunters
Faintest of the Brightest

Shreyas N B[1,2]

[1]Krittika - The Astronomy Club of IIT Bombay, Powai, Mumbai 400076, India

[2] Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

PUBLISHED BY KRITTIKA: THE ASTRONOMY CLUB OF IIT BOMBAY
GITHUB.COM/KRITTIKAIITB
First Release, July 2023

Abstract

Gamma-ray bursts (GRBs) are fascinating astronomical phenomena that have captured the attention of scientists and researchers around the world. These powerful bursts of gamma-ray radiation, lasting from a fraction of a second to a few minutes, provide valuable insights into the universe's workings.

To understand GRBs and their origins, astronomers employ various methods of analysis. In this paper, we will focus on how to analyze data from the Cadmium Zinc Telluride Imager (CZTI) onboard ISRO's famous AstroSat, currently in orbit.

We process the CZTI data using *cztpipeline*, which automatically runs a series of processing modules to yield clean, user-friendly data files. A section in this paper explains the working of the CZTI Pipeline and gives a bird's eye view into building a GUI data processing platform.

Ultimately, we will implement various signal processing concepts on the CZTI data, to mathematically determine a GRB's confidence level.

Contents

I Part One

1 Introduction 7

1.1 Gamma Ray Bursts (GRBs) 7

1.1.1 What are GRBs? 7

1.1.2 Classification of GRBs 8

1.2 AstroSat and CZTI 8

1.2.1 AstroSat - ISRO's first astronomy mission 8

1.2.2 Cadmium Zinc Telluride Imager (CZTI) 9

1.2.3 Instrument Configuration 9

1.3 CZTI Pipeline 11

1.3.1 Working 11

1.3.2 Pipeline Setup and Installation 14

II Part Two

2 Data processing 17

2.1 Introduction 17

2.1.1 FITS file format 17

2.1.2 Using AstroPy 17

2.1.3 Light Curves 18

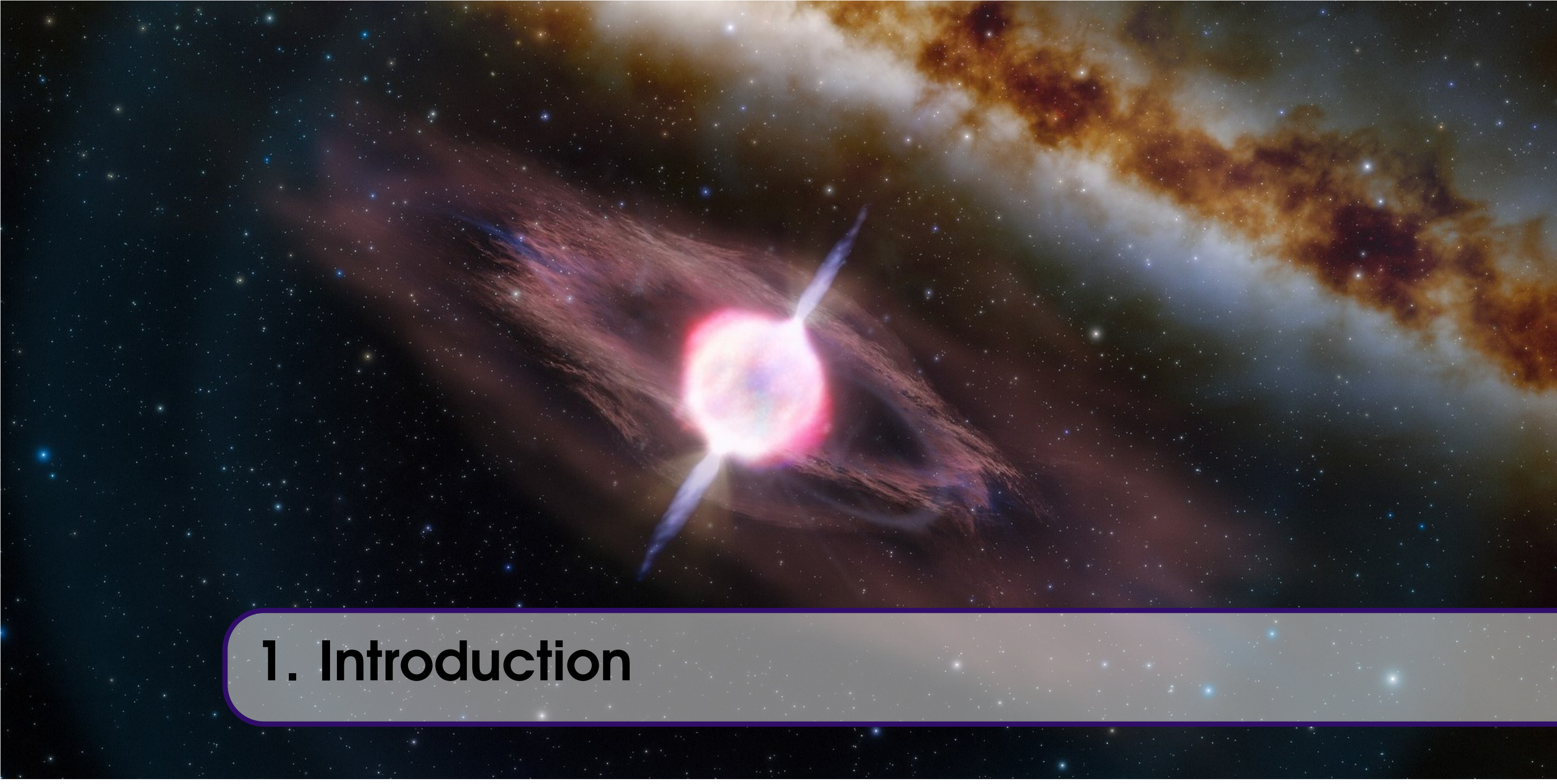
2.1.4 Spectrum and Spectrogram 19

2.1.5 South Atlantic Anomaly 20

2.2 Data Analysis 20
2.2.1 Locating the GRB 20
2.2.2 Filtering and De-Trending 21
2.2.3 SNR Calculations 22
2.2.4 Splitting Energy Bins 25
2.2.5 Outlier Elimination 27
2.2.6 Effect of Time Bins 27
2.2.7 Final Results 29
2.3 Overall Algorithm 35

Part One

1 Introduction 7
1.1 Gamma Ray Bursts (GRBs)
1.2 AstroSat and CZTI
1.3 CZTI Pipeline



1. Introduction

1.1 Gamma Ray Bursts (GRBs)

1.1.1 What are GRBs?

Gamma-ray bursts are highly energetic explosions in distant galaxies which are thought to be released when a supernova implodes to form a neutron star or a black hole. No generally accepted emission mechanism is theorized for a GRB, but some observations indicate that the dominant process may be Inverse Compton scattering. In this model, the pre-existing low-energy photons are scattered by relativistic electrons in the explosion - thus transforming them into gamma rays.

This image by NASA shows that there is also some afterglow emission that occurs (ranging from X-ray to radio) because any energy released by the explosion, not

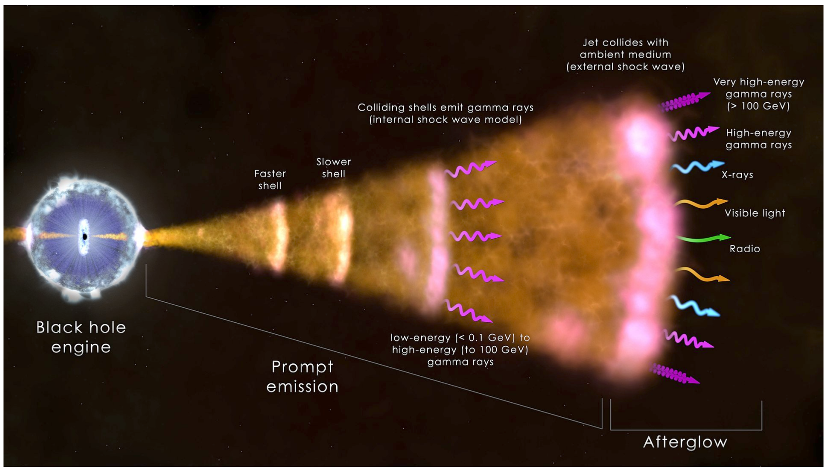

Jet collides with ambient medium (external shock wave)
Very high-energy gamma rays (> 100 GeV)
Colliding shells emit gamma rays (internal shock wave model)
High-energy gamma rays
Faster shell
Slower shell
X-rays
Visible light
Radio
Black hole engine
low-energy (< 0.1 GeV) to high-energy (to 100 GeV) gamma rays
Prompt emission
Afterglow

Figure 1.1: GRB Emission Mechanism

radiated away in the burst, takes the form of matter or energy moving outward at nearly the speed of light. When the matter collides with the surrounding interstellar gas, it propagates a relativistic shock wave into interstellar space.

1.1.2 Classification of GRBs

Short GRBs

Events with a duration of less than 2 seconds. They account for around 30% of all GRBs. The mean time (0.2 seconds) and the fact that the X-ray flashes after a short GRB are observed for a few minutes or hours suggest that short GRBs are probably formed when small particles of a **primary object like a neutron star are initially swallowed by a black hole** in less than two seconds, followed by some hours of lesser energy events.

Long GRBs

Most observed events (70%) have a duration of greater than two seconds and are classified as long gamma-ray bursts. Because these events constitute most of the population, they tend to have the brightest afterglows. Almost every well-studied long GRB has been linked to a galaxy with **rapid star formation**, and in many cases to a **core-collapse supernova** as well, associating long GRBs with the **deaths of massive stars**. Long GRB afterglow observations at high redshift are also consistent with the GRB originating in star-forming regions. In December 2022, astronomers reported the first evidence of a long GRB produced by a **neutron star merger**.

Ultra-long GRBs

These events last more than 10,000 seconds. They have been proposed to form a separate class, caused by the **collapse of a blue supergiant star,** a **tidal disruption event** or a **newborn (magnetar)** Only a small number have been identified to date.

1.2 AstroSat and CZTI

1.2.1 AstroSat - ISRO's first astronomy mission

AstroSat, India's first dedicated multi-wavelength space observatory, stands as a testament to the nation's prowess in space research and exploration. Launched on September 28, 2015, by the Indian Space Research Organization (ISRO), AstroSat has opened new vistas in our understanding of the universe.

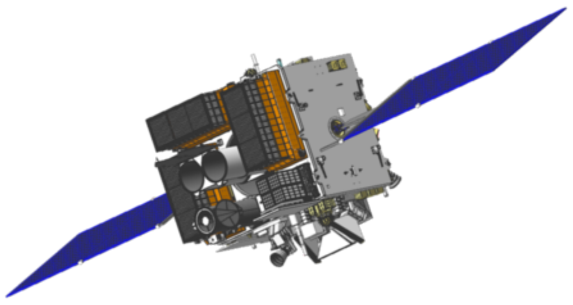


Figure 1.2: AstroSat

One of the key highlights of AstroSat is its ability to observe multiple cosmic sources simultaneously. This capability allows astronomers to conduct comprehensive studies, making significant strides in understanding celestial phenomena. Its primary objectives include the study of distant galaxies, black holes, star-forming regions, and the characteristics of various cosmic sources.

AstroSat has various instruments that operate in multiple wavelengths, including ultraviolet, X-ray, and visible light. This versatile observatory enables scientists to study celestial objects with exceptional precision, unraveling the mysteries of the cosmos. One such famous instrument is the **CZTI**.

1.2.2 Cadmium Zinc Telluride Imager (CZTI)

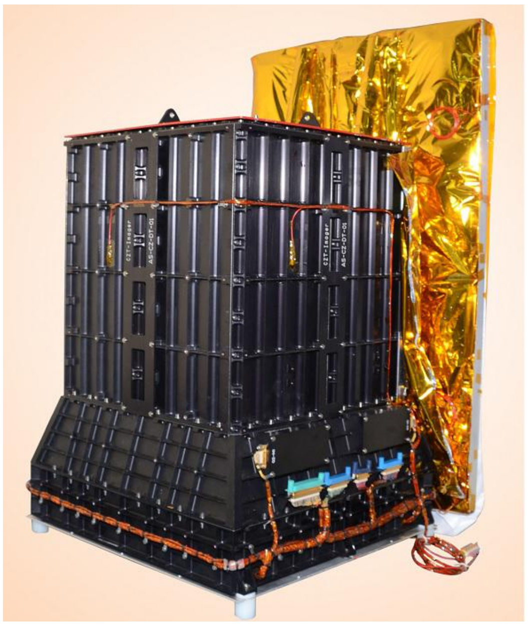


Figure 1.3: CZTI

CZTI (Cadmium Zinc Telluride Imager) is a cutting-edge instrument developed by the Indian Space Research Organization (ISRO) for space-based X-ray astronomy. It is a crucial component of AstroSat, India's first multi-wavelength space observatory.

The Cadmium Zinc Telluride Imager is a high-energy wide-field imaging instrument covering an energy range from 20 to 200 keV and above. It is an open detector above 100 keV, continuously sensitive to GRBs. Unlike previous instruments, CZTI employs Cadmium Zinc Telluride (CdZnTe) detectors, which provide superior energy resolution and efficiency. This enables astronomers to detect and analyze X-rays with exceptional accuracy and sensitivity.

1.2.3 Instrument Configuration

The instrument consists of *four identical independent quadrants*, A–D, to give design safety and redundancy. Each quadrant has a **Coded Aperture Mask** at the top. Sixteen CZT detector modules are mounted 48 cm below low the CAM. Each quadrant also has an alpha–tagged ^{241}Am source for gain calibration just above and to the side of the detector plane, and a CsI (Tl) veto detector below the CZT modules.

Each quadrant has an independent Front end Electronics Board (FEB) for power and signal processing. All four quadrants are connected to common Processing Electronics, which act as the interface to the satellite life. The overall dimensions of the payload are 482 mm × 458 mm × 603.5 mm. In addition to the four quadrants, the instrument has a radiator plate affixed on a side to dissipate heat and provide a cold bias for maintaining CZT module temperatures.

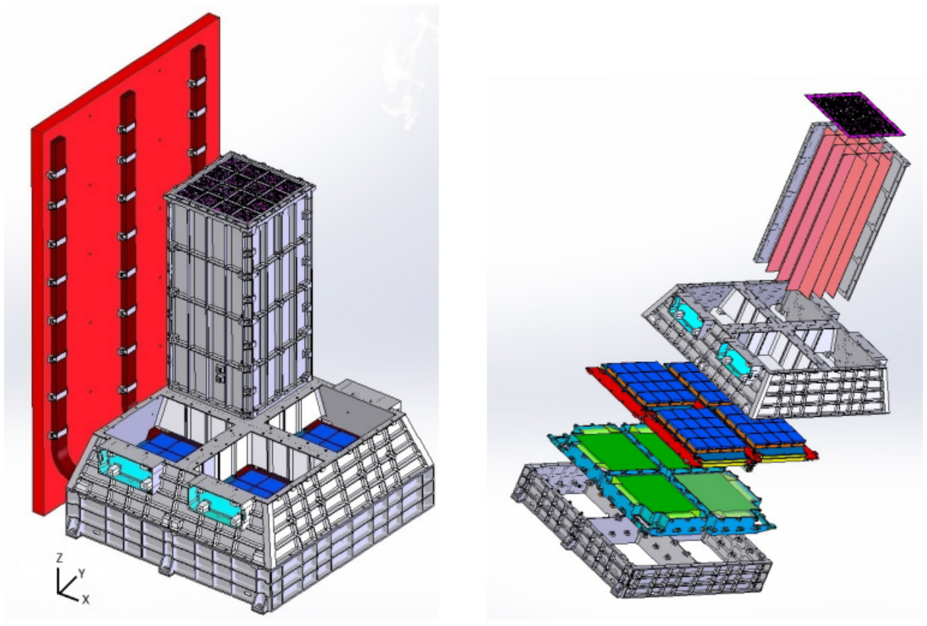


Figure 1.4: Layout of the CZTI

Coded Aperture Mask

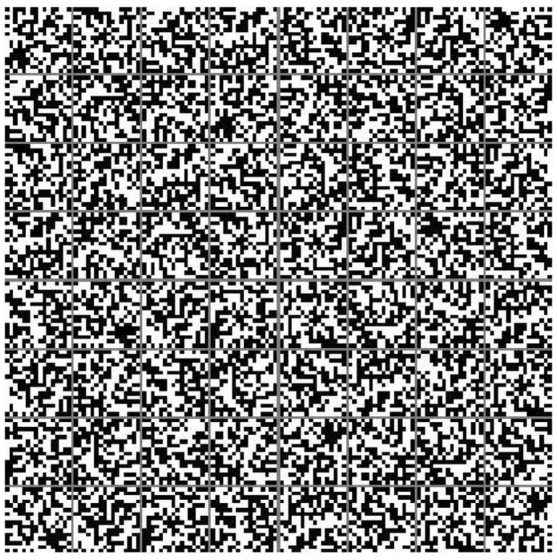


Figure 1.5: Coded Aperture Mask

A Coded Aperture Mask (CAM) is a mask with a pattern of open and closed squares/rectangles which casts a unique shadow on the detector plane for each source direction. The CZTI CAM is made of a 0.5 mm thick Tantalum plate with square and rectangular holes matching the pitch of the CZT detector pixels.

The patterns are based on 255-element pseudo-noise **Hadamard Set Uniformly Redundant Arrays**. 7 of 16 such possible patterns were chosen for individual pixels in the pattern. These seven patterns were placed as a 4 x 4 matrix with some repeats to generate the CAM. This pattern is placed on other quadrants, rotated clockwise by 90°, 180°, and 270°, respectively.

1.3 CZTI Pipeline

To summarize the instrumentation, the CZTI is configured as four independent quadrants, each with 16 detector modules. Each detector module has 256 pixels. Collimator slats surround every detector module, and the Coded Aperture Mask is placed above the collimator.
The instrument operates in event mode, wherein the time, energy, and pixel coordinates of each photon event are recorded in the data. There is a CsI Veto detector under the CZT modules. If an event recorded in the Veto occurs within a specific short time window of any CZT event, the latter is tagged with a Veto flag. The instrument also contains Alpha-tagged calibration sources, which generate 60 keV photons simultaneously with an Alpha particle. CZT events within a defined time window of alpha detection are marked with an Alpha tag in the recorded data.

1.3.1 Working

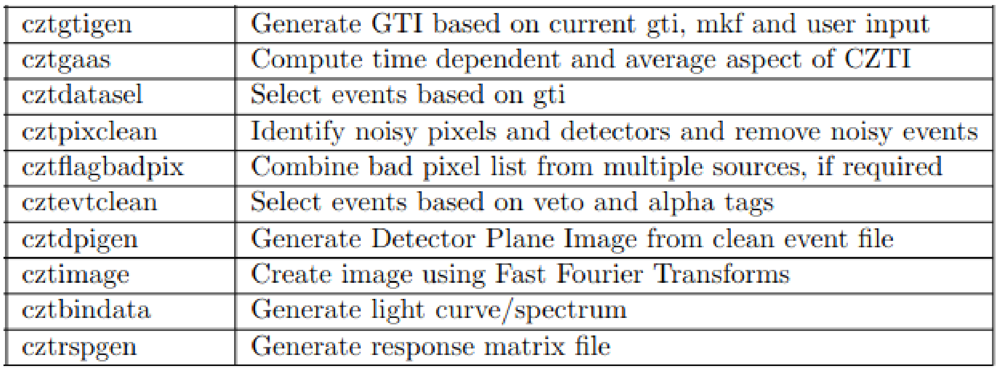
The data received from the CZTI payload passes through a series of steps in a processing pipeline. Three main data levels have been designated for long-term storage. These are:

Level 0
This is the raw data received from satellite telemetry, which is segregated by instrument and auxiliary data. This data is archived internally and not distributed for public use.

Level 1
This is reorganized raw data, written in FITS format for Astronomical use. All the auxiliary information necessary for further processing this data is collated at this level and packed with the respective science data. This data is released via AstroSat data archive to the Principal Investigator (PI) of the corresponding observing proposal and, after a specified initial lock-in period, to anyone interested in the data.

Level 2
This data contains standard science products derived from Level 1 data. Level 2 data is also in FITS format and is available for science use, with the same lock-in criteria and release mechanism as the Level 1 data. The software to process data from Level 1 to Level 2 contains user-configurable elements. While a default configuration is run at the Payload Operation Center (POC) to create the automated Level 2 standard data products, the user can generate more customized products using the same software with other configuration settings. The Level 2 pipeline software is released publicly for general use.

The CZTI level-2 data reduction pipeline takes the dataset generated at Level 1 as input. The original data is received via telemetry at the Indian Space Science Data Center (ISSDC) during each visible pass over Bengaluru, India. Subsequently, this data is packaged during each download orbit and transmitted to the Payload Operations Center (POC) for additional processing. At the POC, all the data associated with a specific Observation ID **(ObsID)** is collected from different orbit-based packages, resulting in a merged Level 1 product. This merged product



cztgtigen	Generate GTI based on current gti, mkf and user input
cztgaas	Compute time dependent and average aspect of CZTI
cztdatasel	Select events based on gti
cztpixclean	Identify noisy pixels and detectors and remove noisy events
cztflagbadpix	Combine bad pixel list from multiple sources, if required
cztevtclean	Select events based on veto and alpha tags
cztdpigen	Generate Detector Plane Image from clean event file
cztimage	Create image using Fast Fourier Transforms
cztbindata	Generate light curve/spectrum
cztrspgen	Generate response matrix file

Figure 1.6: CZT data processing modules for Level 2 data

is then created for distribution and further analysis. The Level 2 pipeline analysis procedure, outlined below, begins with the merged Level 1 package, which comprises the following files:

1. **Science Data File**: a **FITS** file containing sequentially stacked 2048-byte data frames generated by the CZTI payload. This data is mode segregated, i.e., a different FITS file is generated for each distinct data mode.
2. **Time Calibration Table**(.tct): It includes a compilation of time tags presented in the CZTI internal clock, Satellite On Board reference Time (OBT), and Universal Time (UT) derived from the SPS units on AstroSat. Since the CZTI science data is labeled with its internal clock, the TCT file is essential for establishing correlations with other onboard events and obtaining accurate timing information.
3. **Orbit File**(.orb): Gives time-tagged orbital position of the satellite in terms of geocentric x, y, z as a function of time, as well as the corresponding velocity components.
4. **Attitude File**(.att): Gives time-tagged satellite attitude information in terms of quaternions as well as the RA and Dec values of the pointing directions of the three reference axes of the satellite.
5. **Make Filter File**(.mkf): This file collects together the time series of several selected parameters, including health parameters, Sun angle, Earth angle, charged particle count, etc., based on which the quality of data obtained can be assessed.
6. **LBT Housekeeping File**(.lbt): This gives a time-tagged recording of 65 different health parameters monitored by different sensors on the CZTI.
7. **MCAP File**(.mcap.xml): This XML file has information like source observed, start and end time of observation etc

We shall be directly working with level 2 data, i.e., starting with the Event file (.evt), Make filter file (.mkf), the Make Filter Thresholds file (mkfThresholds.txt) and the Livetime file (.fits).

If you can access preliminary level 2 data, you can always run the required modules to generate the above files. The data processing modules can be individually run in the order shown in Figure 1.6.

The Figure 1.7 gives a brief idea on how the pipeline's workflow is. However, it is advised to refer to CZTI Pipeline Guide thoroughly for an understanding of how the above data reduction modules work.

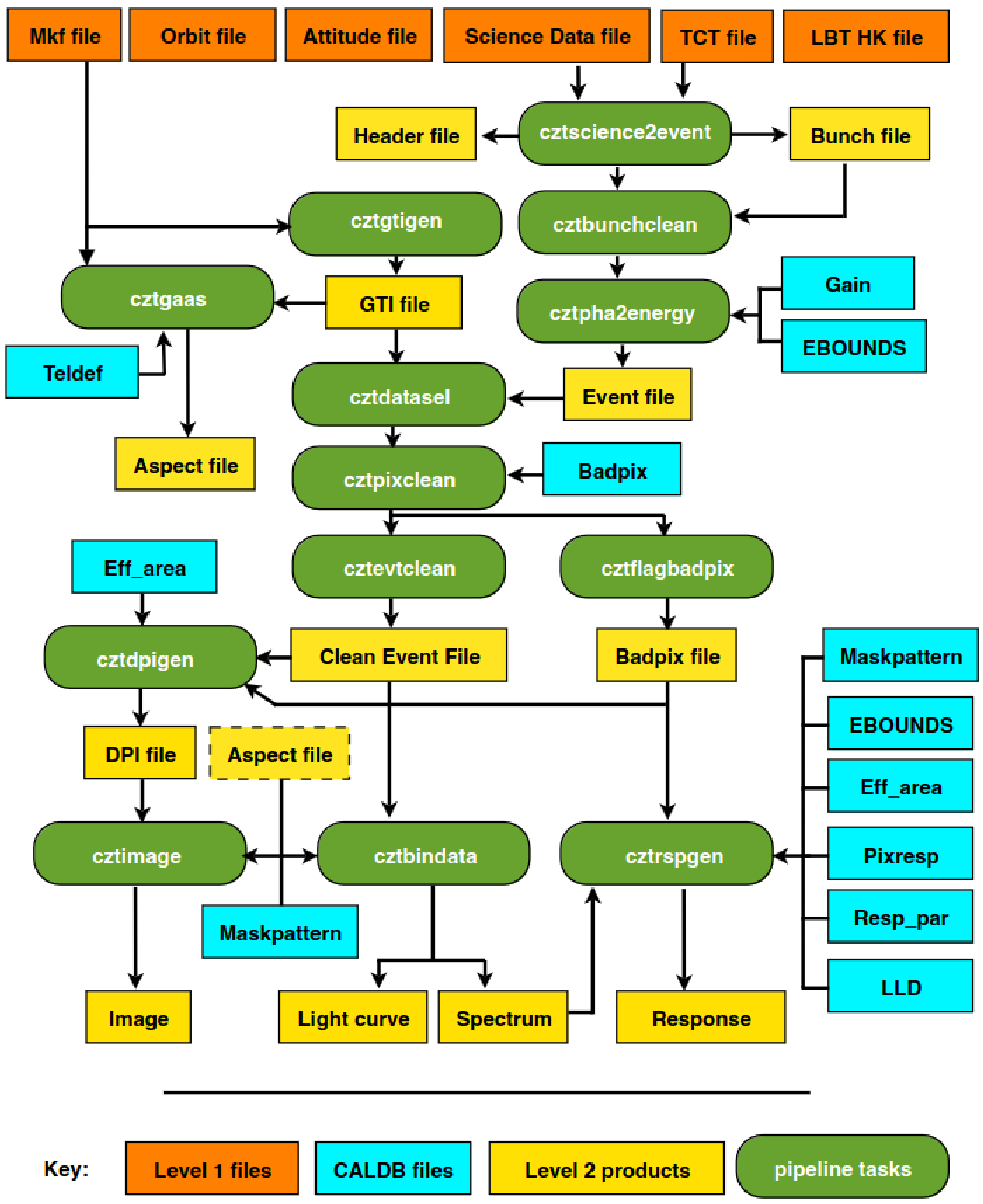


Figure 1.7: Workflow of the Pipeline

1.3.2 Pipeline Setup and Installation

I have set up the pipeline on **Ubuntu 20.04 Dual Boot**. You can use the same version as WSL too. Higher versions (or other distros) may have compatibility issues with the C++ compiler. The steps for installation are as follows:
Start by downloading the pipeline from - AstroSat SSC. Untar it by using

```
$ tar -xvf czti_pipeline_20221209_v3.0.tar
```

Also install the CALDB file from here. Untar this too by using

```
$ tar -xvf czti_caldb_20221209.tar
```

Before proceeding, it is always better to run

```
$ sudo apt update && sudo apt upgrade
```

Now install the prerequisites gcc (GCC 4.4+), gfortran, and Perl by using

```
$ sudo apt install gcc g++ gfortran perl make
```

Now run

```
$ nano ~/.bashrc
```

and add the following lines to the file:

```
export as1czt=home/user/Downloads/czti_pipeline/czti
export PFILES=$as1czt/paramfiles
export PATH=$as1czt/bin:$as1czt/scripts:$PATH
export GLOG_log_dir=$as1czt/log
export CZTI_templates=$as1czt/templates
export PERL5LIB="$as1czt/lib/":$PERL5LIB
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$as1czt/lib
ulimit -s 65532
export CALDB=home/user/Downloads/data/as1/czti
```

Note that you need to put your own path in the above code.
Finally, run the modified file and install libraries by executing the following:

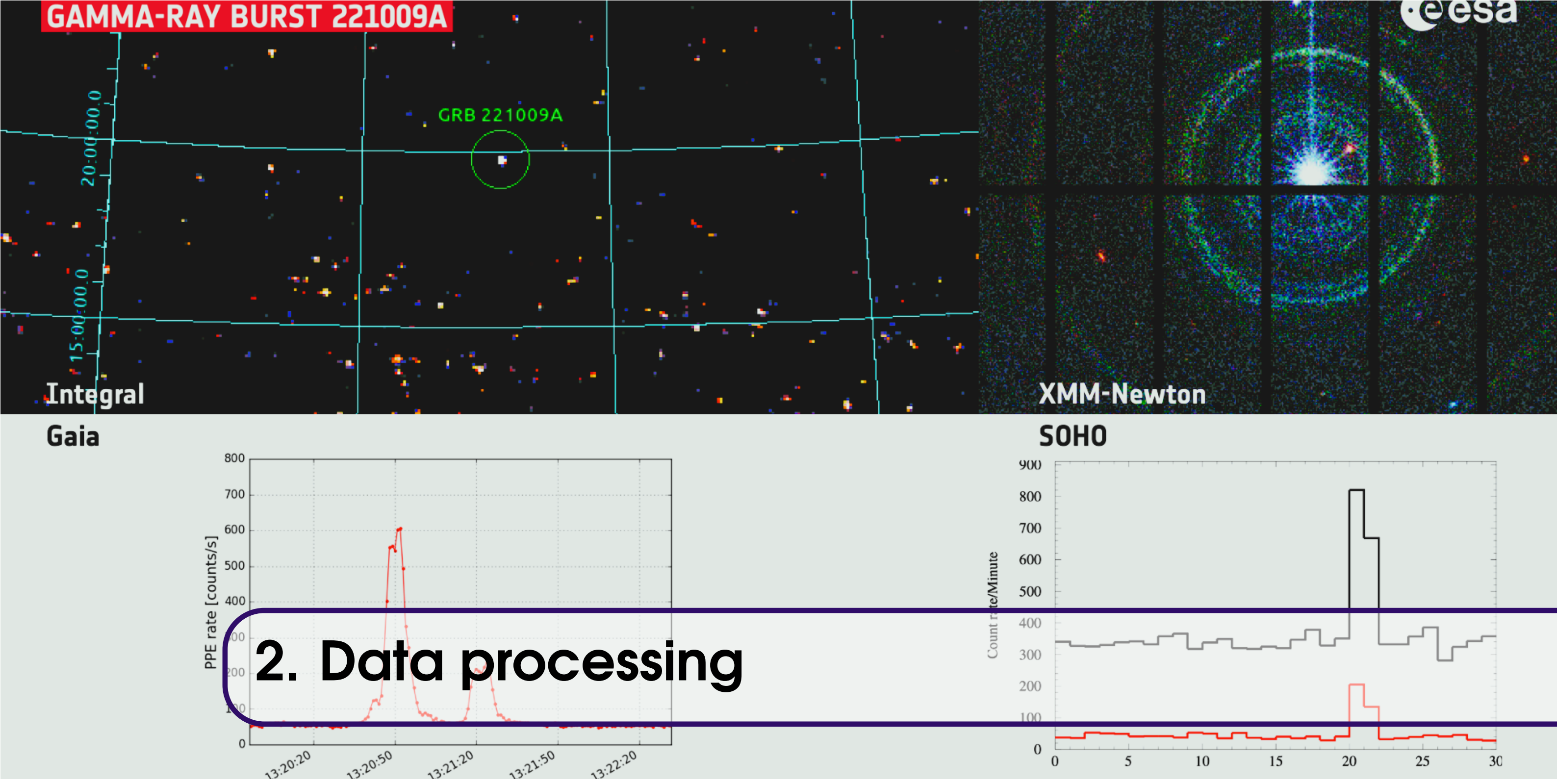
```
$ source ~/.bashrc
$ cd $as1czt
$ cd ../
$ ./InstallLibs
$ cd $as1czt
$ make
$ cd scripts
$ chmod +x cztpipeline
```

Your CZTI pipeline should now be ready and working. You can check by running

```
$ cztpipeline
```

Part Two

2 Data processing 17
2.1 Introduction
2.2 Data Analysis
2.3 Overall Algorithm



2. Data processing

2.1 Introduction

In this chapter, we look at visually analyzing GRB data processed using the CZTI Pipeline. Any results mentioned in this section refer to the processed data of **GRB190928A**. But before diving into the actual code, let us understand a few basic concepts of data storage and data analysis in the context of astronomy.

2.1.1 FITS file format

The **Flexible Image Transport System** (FITS) format is a widely adopted standard in astronomy for storing, transmitting, and processing scientific data. FITS was designed to address the specific needs of astronomical data, accommodating a range of data types, including **images, spectra, and tables**.

FITS files have two primary components: the **data unit (HDU)** and the **header**. The HDU contains the actual data, organized in multidimensional arrays, while the header stores metadata describing the data's origin, properties, and processing history. This self-describing nature of FITS files ensures that crucial information accompanies the data, enabling interoperability and facilitating reproducible research.

2.1.2 Using AstroPy

AstroPy, a prominent open-source library for astronomers, offers a range of functionalities, including the ability to open and manipulate FITS files. With AstroPy, opening FITS files becomes effortless, providing a simple and intuitive interface for reading, writing, and manipulating such files. The library enables users to access header information, extract image data, and perform various operations like data manipulation, visualization, and analysis.
Let us try opening a FITS file using AstroPy.

```python
from astropy.io import fits
hdul = fits.open('data.fits')
hdul.info()
```

This will print the information about the FITS file - such as the HDUs in it, the name, dimensions, etc., of each extension. To get the FITS file's data and header, we do:

```python
data, header = hdul[0].data, hdul[0].header
```

2.1.3 Light Curves

What is a Lightcurve?

The lightcurves of gamma-ray bursts (GRBs) are vital tools for understanding the nature and properties of these energetic cosmic events. A lightcurve is a graph that displays the variation of brightness over time. For GRBs, lightcurves capture the intense emission of gamma rays followed by the afterglow across various wavelengths. By analyzing the shape, duration, and spectral characteristics of GRB lightcurves, we can gain insights into the physics behind these powerful phenomena.

We can create lightcurve files using the `cztbindata` module, which takes in the following inputs:

- `evtfile` - input event file
- `mkffile` - input make filter file
- `badpixfile` - input bad pixels file
- `livetimefile` - input livetime file
- `time bin size` - $1s$ by default

The module produces the lightcurve (*.lc*) files and spectrum (*.pha*) files for all 4 quadrants. These files can also be opened using AstroPy. We can plot the lightcurve in python:

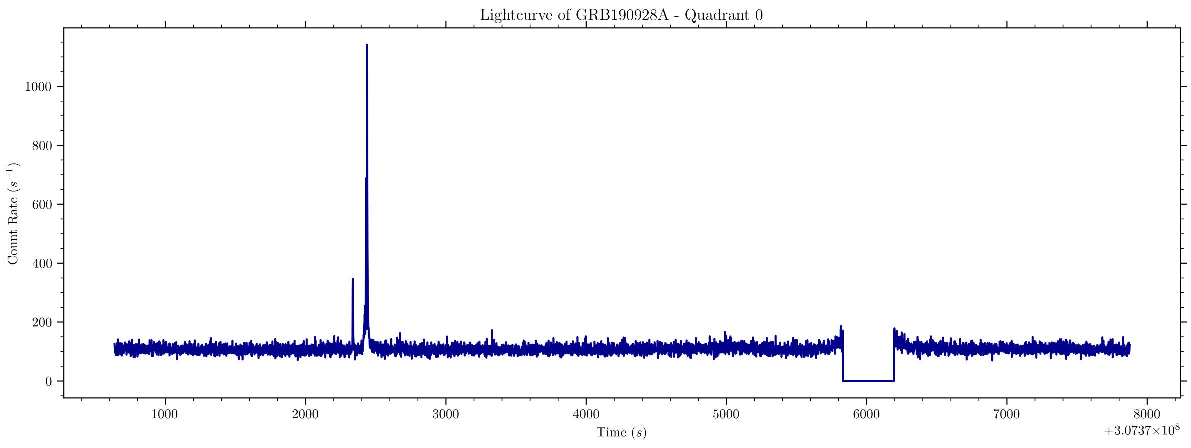


Figure 2.1: Lightcurve of GRB190928A

2.1.4 Spectrum and Spectrogram

Spectral plots visualize spectral flux against wavelength or, equivalently, photon count versus energy. We have used `cztbindata` to get the spectrum data in the form of (*.pha*) file. Plotting the `CHANNEL` versus `COUNTS` for all four quadrants looks like:

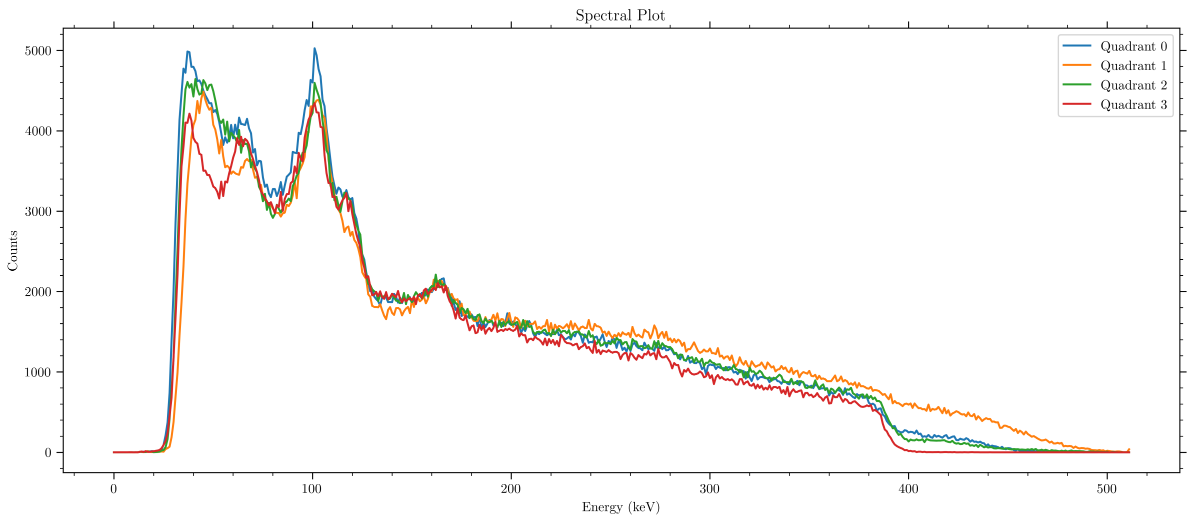


Figure 2.2: Spectrum Plots of GRB190928A

We can also plot a spectrogram, which is a visual representation of the spectrum of frequencies of a signal as it varies with time. It looks like this:

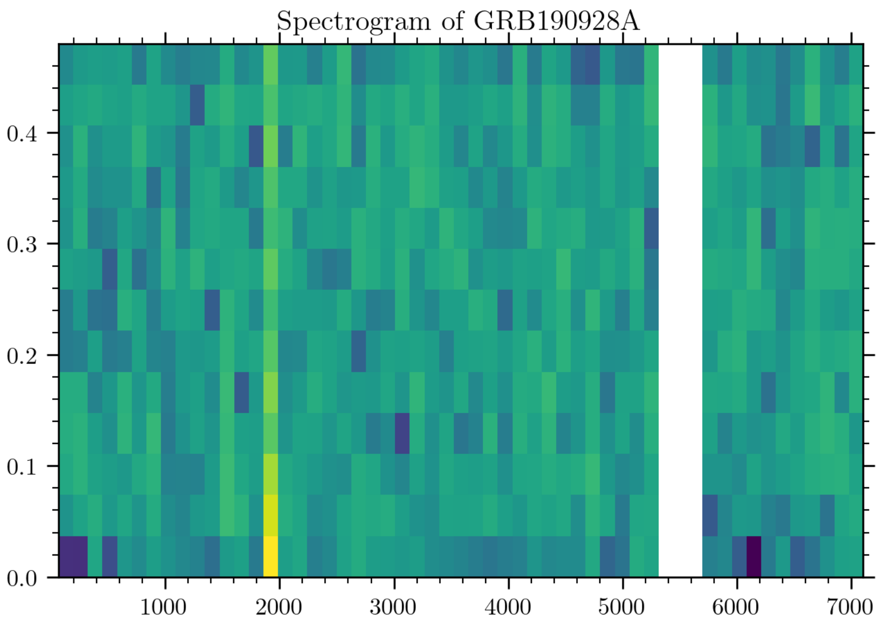


Figure 2.3: Spectrogram of GRB190928A

2.1.5 South Atlantic Anomaly

The **South Atlantic Anomaly** (SAA) is a region of space located over the South Atlantic Ocean where the Earth's inner Van Allen radiation belts come closest to the Earth's surface. This anomaly poses challenges for satellites and spacecraft passing through this region due to increased radiation levels.
The SAA is caused by the interaction of the Earth's magnetic field with high-energy particles from the Sun, known as the solar wind. The Earth's magnetic field typically shields the planet's surface from these particles. However, the configuration of the magnetic field in the South Atlantic region allows the radiation belts to come closer to the Earth's surface, exposing spacecraft to higher radiation levels.
This is why you may have noticed a **zero count rate region** around the 6000 timestamps because the CZTI is turned off when it passes over the SAA. It is important to

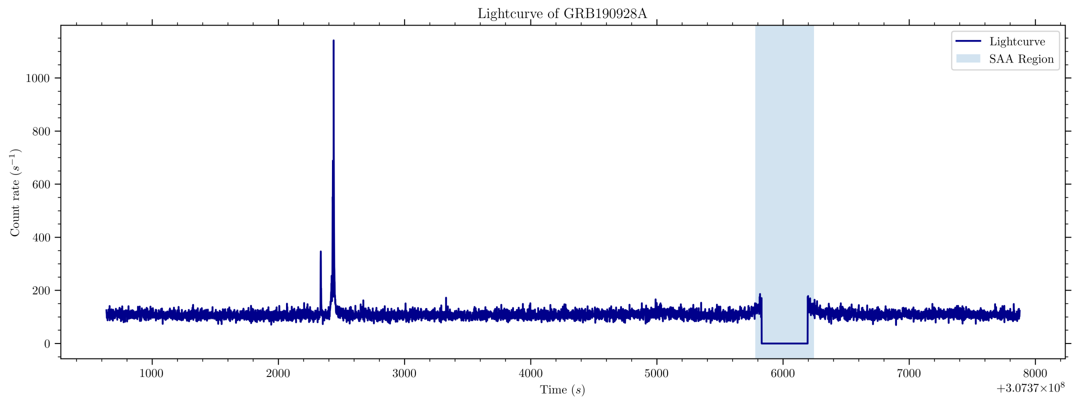


Figure 2.4: Lightcurve with SAA Region highlighted

understand that we should not include such regions of anomaly while processing or fitting the data.

2.2 Data Analysis

2.2.1 Locating the GRB

Let us plot the lightcurve of GRB160909A and try to analyze it visually.

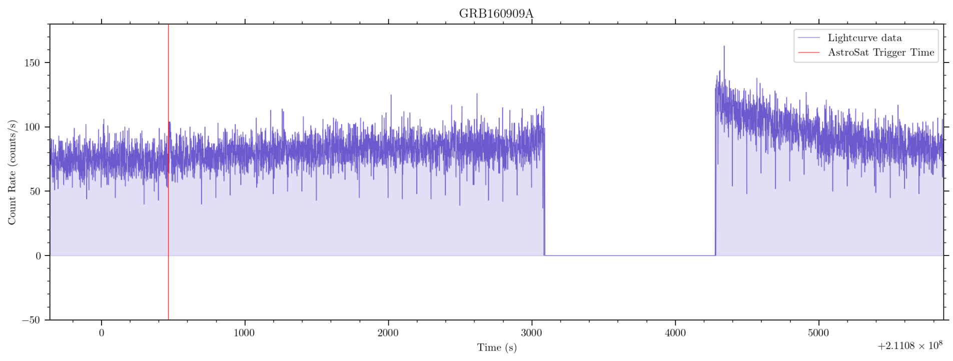


Figure 2.5: Lightcurve of GRB160909A

From the AstroSat GRB Monitor Page, we get a (possible) trigger time for each GRB. In the above plot, the trigger time of 211080464.0 is shown. This means that the GRB *could* lie here. With this information, we perform filtering and de-trending by choosing a window before and after the trigger time. This process is discussed in detail in the upcoming sections. Our objective would be to use this window to process the data from different energy bins, calculate and compare SNRs across time bins, eliminate outliers, and finally output if there was indeed a GRB found at the trigger time, which AstroSat gives.

2.2.2 Filtering and De-Trending

Looking at the lightcurve, you may have noticed that the background data is around 80 counts, and it follows a *trend*. This can be due to various reasons, and to correctly analyze the signal and the corresponding SNR later, we need to remove this trend from the background data. We can use a **running filter** on the background data. We use a **Savgol filter** of $100s$ window with a second-order polynomial and then fit a quadratic polynomial to this filter. To fit the filter and to perform de-trending, we can consider a window of $500s$ before and after the trigger time. This gives us the following plot:

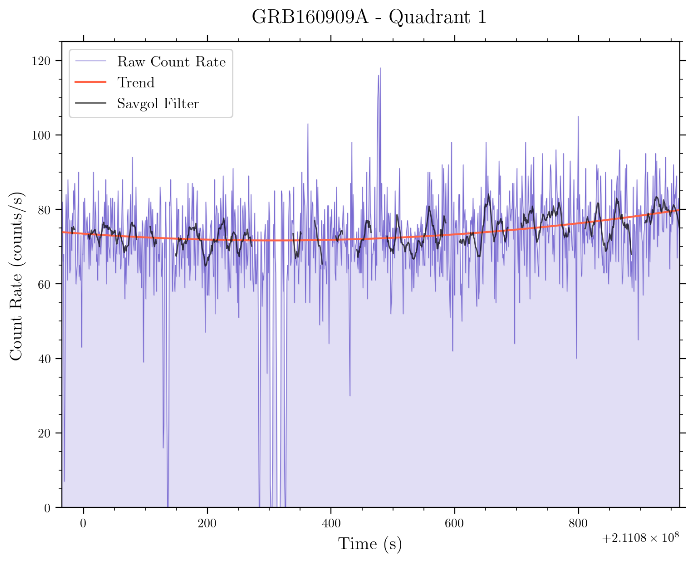


Figure 2.6: Identifying the Trend in the Lightcurve

We subtract the quadratic polynomial from the entire GRB window of $1000s$ to get the detrended GRB window. This looks like the following:

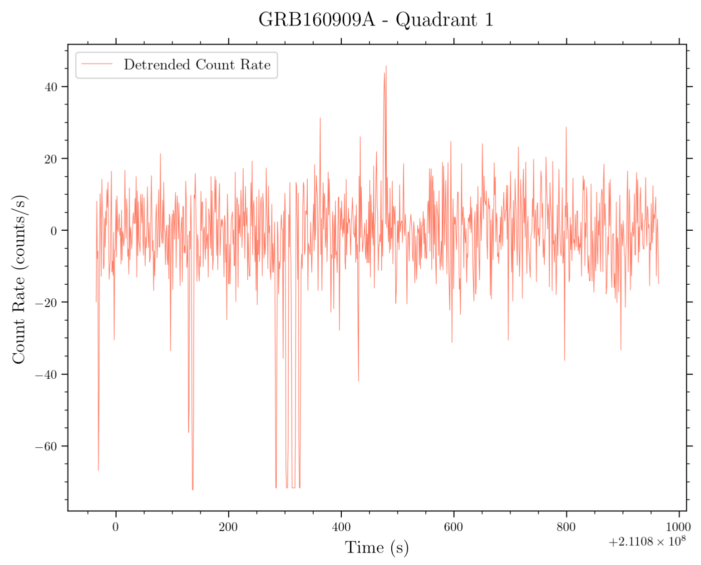


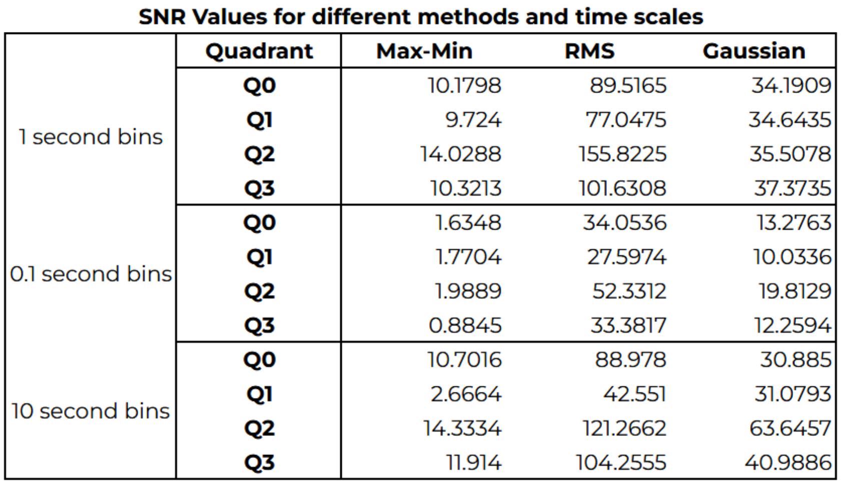
Figure 2.7: Detrended Lightcurve

As we can see, the detrended light curve now has a median noise of zero, and this amplitude of the GRB corresponds to the actual intensity. We also exclude the SAA region from detrending as the CZTI would be shut down during this time. This means we need to check if the GRB window contains the SAA region during detrending. If it does, we need to exclude the SAA region from detrending.

2.2.3 SNR Calculations

vital tool in analyzing any kind of signal is the **Signal-to-Noise Ratio** (SNR). There is no fixed method to calculate the SNR of any signal. We implement the following models and choose the one which model makes more sense physically and mathematically:

1. **Noise Max-Min fitting**: We calculate the (maximum-minimum) of the noise and consider this as the noise threshold. We say that the SNR is equal to the peak value of the GRB divided by the noise threshold.
2. **Noise RMS fitting**: We calculate the RMS value of the noise and say that this is the noise threshold. In this way, SNR would be the peak value of GRB divided by the RMS value of noise.
3. Noise Gaussian fitting: We fit a Gaussian function to the histogram of background data and say that the noise threshold is 3 times the standard deviation above the mean of this Gaussian function. Thus, SNR would be the peak of GRB divided by this noise threshold.

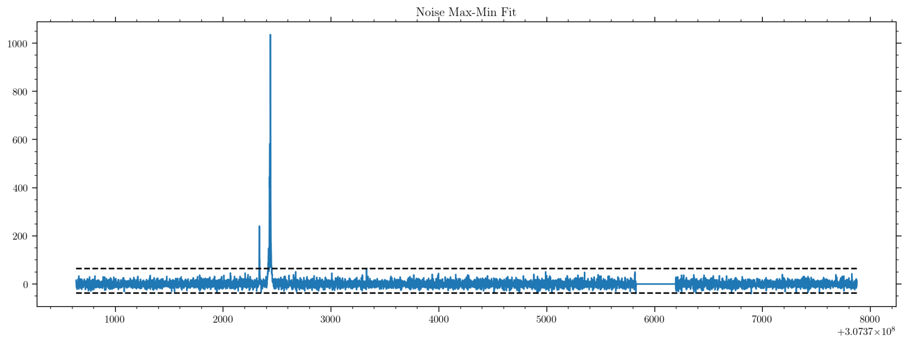


SNR Values for different methods and time scales

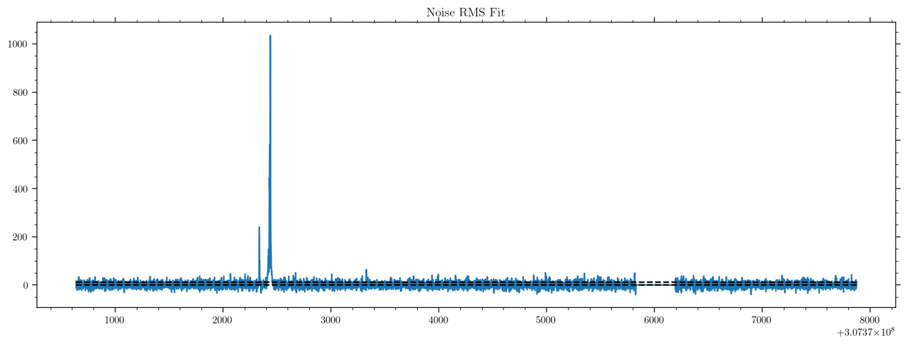
	Quadrant	Max-Min	RMS	Gaussian
1 second bins	**Q0**	10.1798	89.5165	34.1909
	Q1	9.724	77.0475	34.6435
	Q2	14.0288	155.8225	35.5078
	Q3	10.3213	101.6308	37.3735
0.1 second bins	**Q0**	1.6348	34.0536	13.2763
	Q1	1.7704	27.5974	10.0336
	Q2	1.9889	52.3312	19.8129
	Q3	0.8845	33.3817	12.2594
10 second bins	**Q0**	10.7016	88.978	30.885
	Q1	2.6664	42.551	31.0793
	Q2	14.3334	121.2662	63.6457
	Q3	11.914	104.2555	40.9886

Figure 2.8: SNR Results

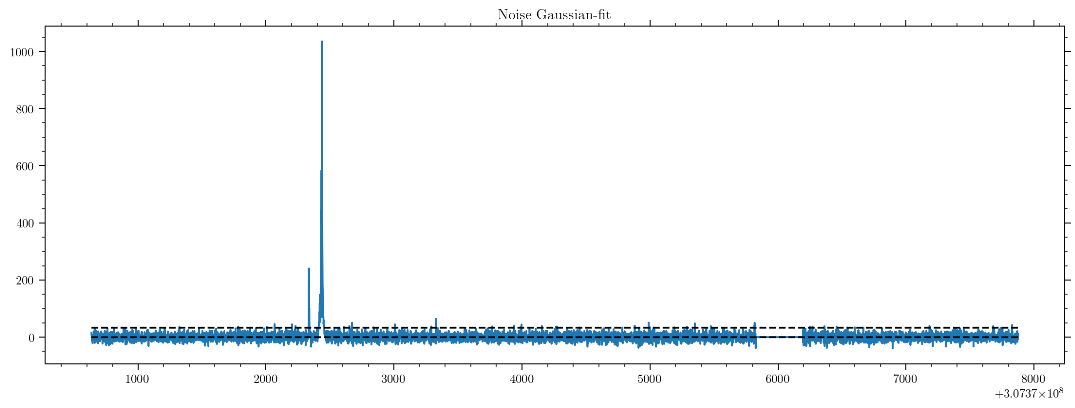
Let us try explaining which method is better from the above results.



As we can tell, the Max-Min fit overestimates the noise, resulting in comparatively low values for SNR. We can see this in the SNR Results too. Let's look at the RMS noise threshold.



From the above, we can tell that RMS fit under-estimates the noise, resulting in comparatively high values for SNR. Now, the Gaussian fit results in a noise threshold



that seems like a reasonable fit of the noise, thus resulting in "good" values for SNR. Let us see how the Gaussian fits the background data.

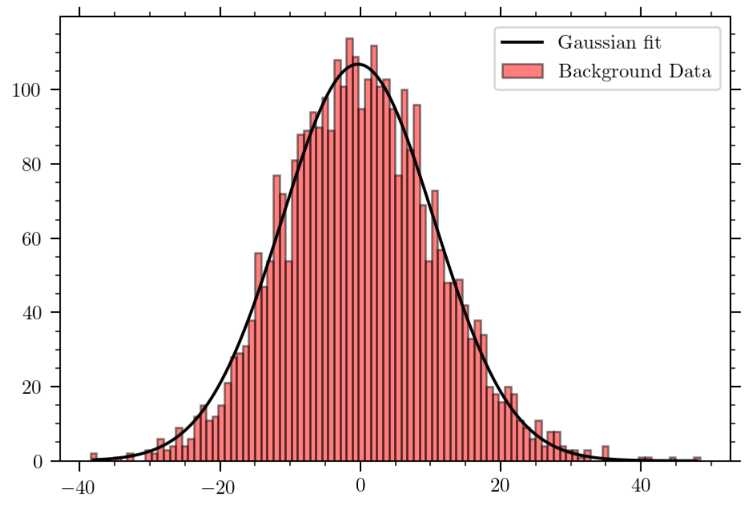


Figure 2.9: Noise Gaussian fitting of background data

Thus, we can see that since the noise is not a Gaussian distribution, the best thing to do is not fit any model and apply `sigma_clipped_stats`, which gives the mean, median and standard deviation of the lightcurve, which can be used to eliminate noise by choosing an optimal multiple of the standard deviation (3 by default).

2.2.4 Splitting Energy Bins

We can generate light curves for different energy bins. The total energy range of a lightcurve by default is $20-200keV$. We split this into energy bins to identify and exclude outliers. We need to choose an optimal number of energy bands because choosing too many bins decreases the information in each bin, and using too few bins will not allow us to compare and differentiate energy characteristics. Let us consider the following energy bins:

- 2 bins
- 3 bins
- 4 bins
- 6 bins
- 8 bins
- 10 bins

Let us consider the GRB190928A. Using `cztbindata`, we generate lightcurves for each of the above n energy bins (total $\sum n = 33$ lightcurves). In each case, we detrend the lightcurves and calculate the SNRs of the outliers, which have a count rate greater than 3σ than the mean. Here, the mean and standard deviation (σ) are obtained by fitting a Gaussian to the background data discussed in the previous section. Now, for each n bin, we extract the maximum SNR, as we can safely assume that this corresponds to the GRB itself. We plot this across the energy ranges to see how the SNR changes across energy bins for each n. We can also plot the signal (peak rate) across energy bins. This is what we get:

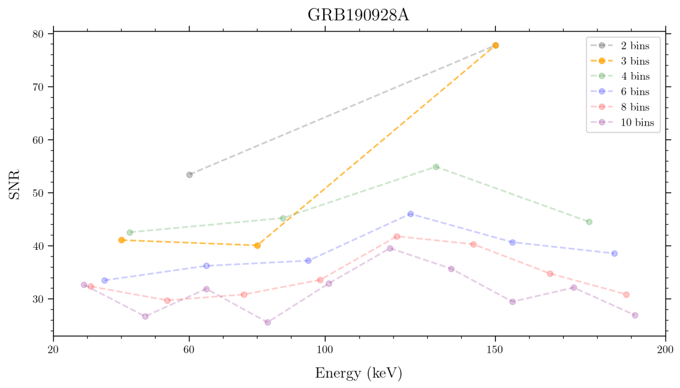


Figure 2.10: SNR comparison across energy bins

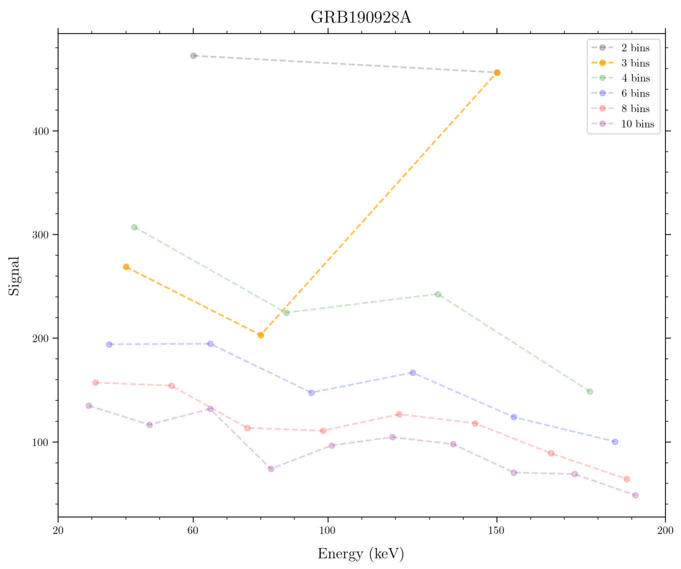


Figure 2.11: Signal comparison across energy bins

From the above plots, the 3 energy bins trend looks conspicuous. The SNR and signal plots show an initial decrease in their respective values from the $20-60keV$ band to the $60-100keV$ band, followed by a drastic increase when we move from the $60-100keV$ band to the $100-200keV$ band. On analyzing these bands from an astronomical point-of-view, it makes sense to take 3 energy bins because we usually encounter outliers like cosmic rays and others in the $20-60keV$ band (sometimes also taken as the $20-50keV$ band); we encounter phosphorous fluorescence in the $60-100keV$ band; and we generally expect the gamma-ray burst to be in the $100-200keV$ band.
Thus, we consider the 3 energy band split for further analysis.

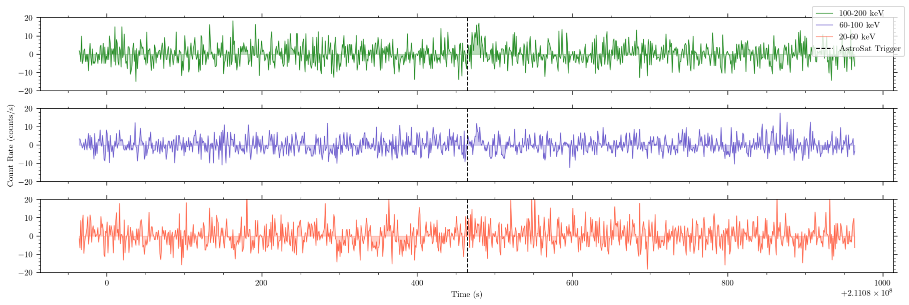


Figure 2.12: Lightcurves for the 3 energy bands

2.2.5 Outlier Elimination

Since we know that the $20-60keV$ energy band usually contains most of the outliers, we can find the indices of these outliers in this band and compare the SNRs for the outliers at these indices across energy bands. By doing this, we expect that the SNR of the outliers decreases as we move to higher energy bins, and only the SNR corresponding to the GRB must increase. We can plot these 3 masked lightcurves on top of each other to compare the SNRs across energy bins for a single quadrant:

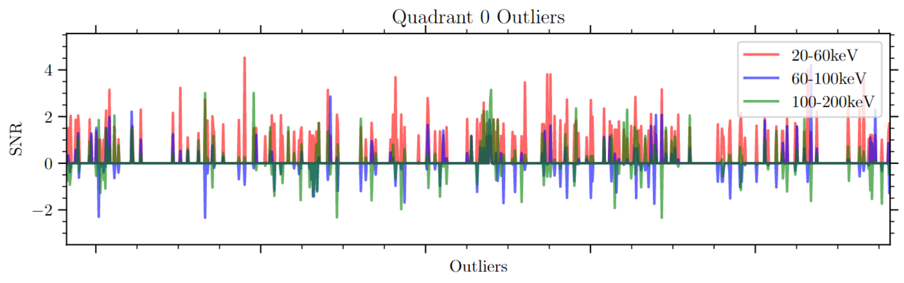


Figure 2.13: SNRs of outliers from lower energy band across energy bins

In the above plot, we can see that the SNRs of the outliers, which were detected in the $20-60keV$ energy band, decrease as we move to the higher energy bands. But the SNRs of the potential GRBs increase and remain considerably above $SNR = 1$ in the higher energy bands. This is how we differentiate an outlier (like cosmic rays) from a gamma-ray burst. Eliminating the outliers and showing only the potential GRBs, we are left with the following plot for a single quadrant:

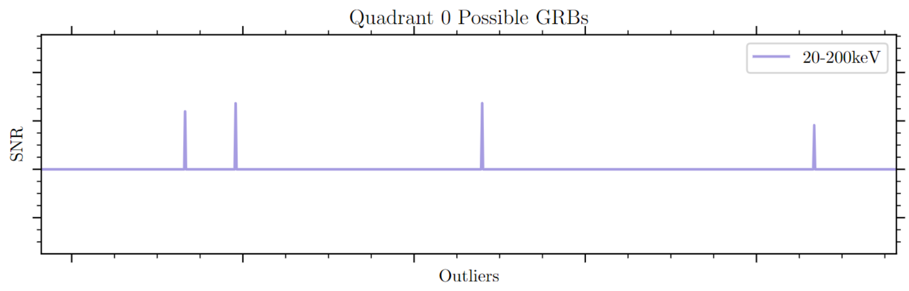


Figure 2.14: Potential GRBs in $20-200keV$ range

The next level of outlier elimination will be done by cross-matching against different quadrants. This is done considering that a gamma-ray burst will have the same time stamps across multiple quadrants.

2.2.6 Effect of Time Bins

As we already know, gamma-ray bursts have various periods ranging from $0.01s$ to $100s$, and we need to choose an optimal time bin size to record the GRB counts efficiently. Taking a time bin too small will result in higher noise variation, lowering the SNR value. At the same time, if we choose a time bin too large, the GRB (signal) peak is lowered since the counts are getting added in each time bin. This implies that the value of the signal decreases, hence also decreasing the SNR value. Thus, choosing an optimal time bin size is crucial. By default, `cztbindata` generates

lightcurves of bin size, $1s$, as it works well for most cases. However, it might not be the time bin size for which we record the GRB peak because if the GRB period is $0.5s$, for example, sampling every $1s$ may miss this peak count rate, resulting in wrong SNR calculations. Thus, let us generate lightcurves of different time bin sizes (in seconds) - $0.2, 0.4, 0.8, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0,$. We can process these lightcurves to calculate and compare the SNRs of the GRB detected.
Following the above processing procedures, we can plot SNRs versus time-bins:

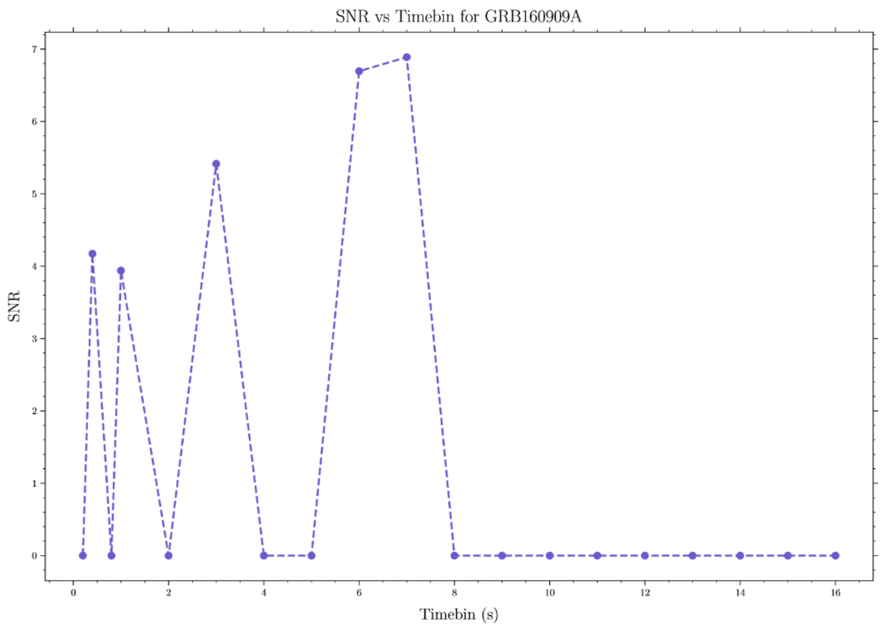


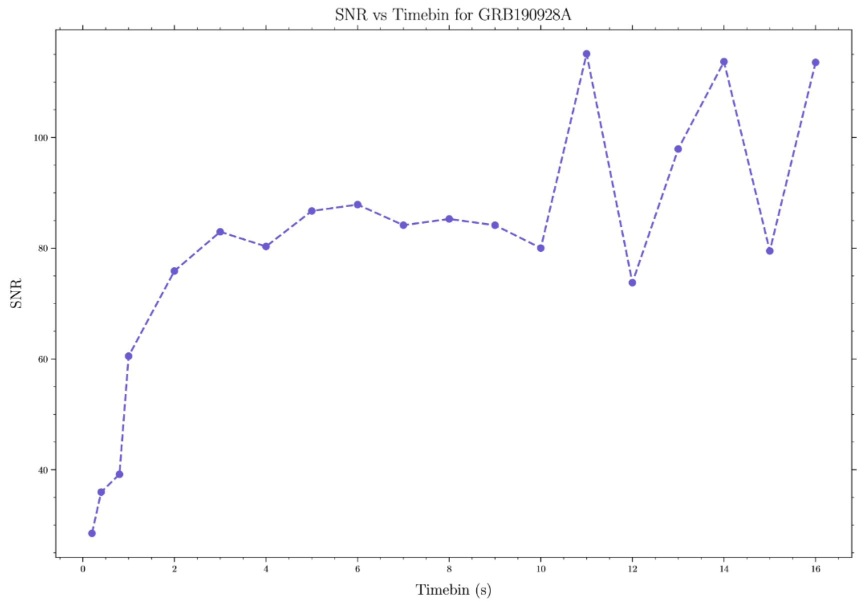
Figure 2.15: SNR of the Potential GRB versus time bin sizes

Thus, we can see that for a time bin size of $7s$, this particular GRB will yield maximum SNR. So it makes sense to perform our final analysis on a $7s$ time-binned lightcurve. Results are shown in the next section.

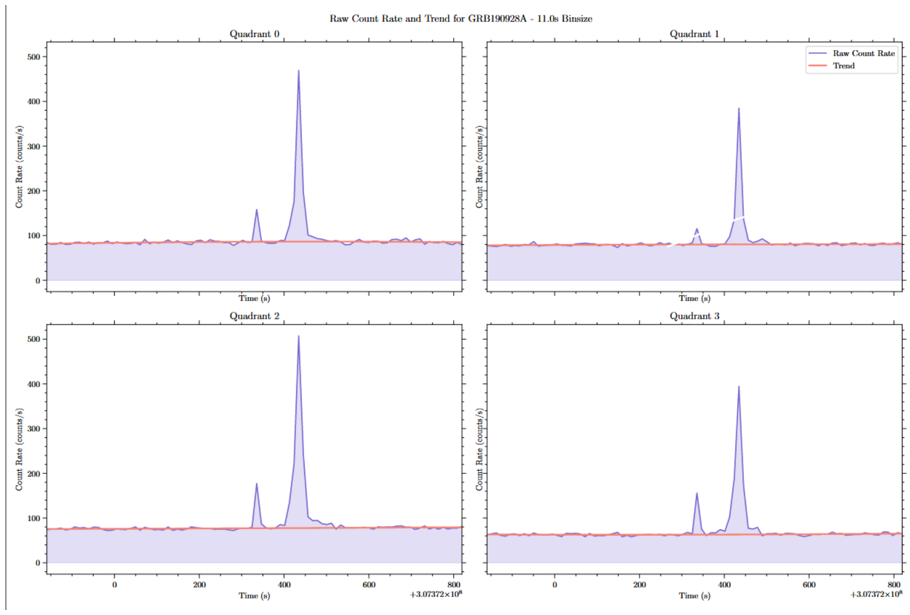
2.2.7 Final Results

GRB 190928A

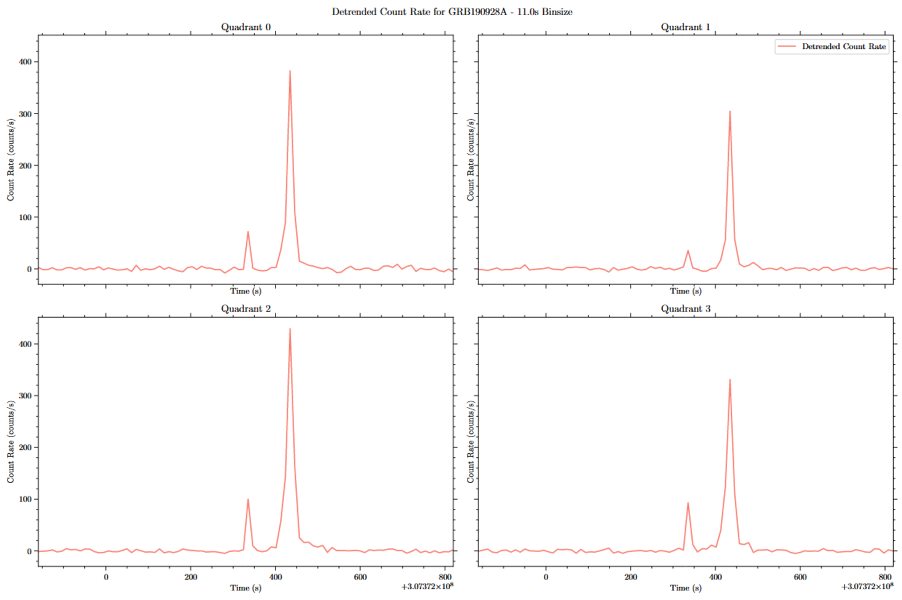
SNR versus time bins sizes:



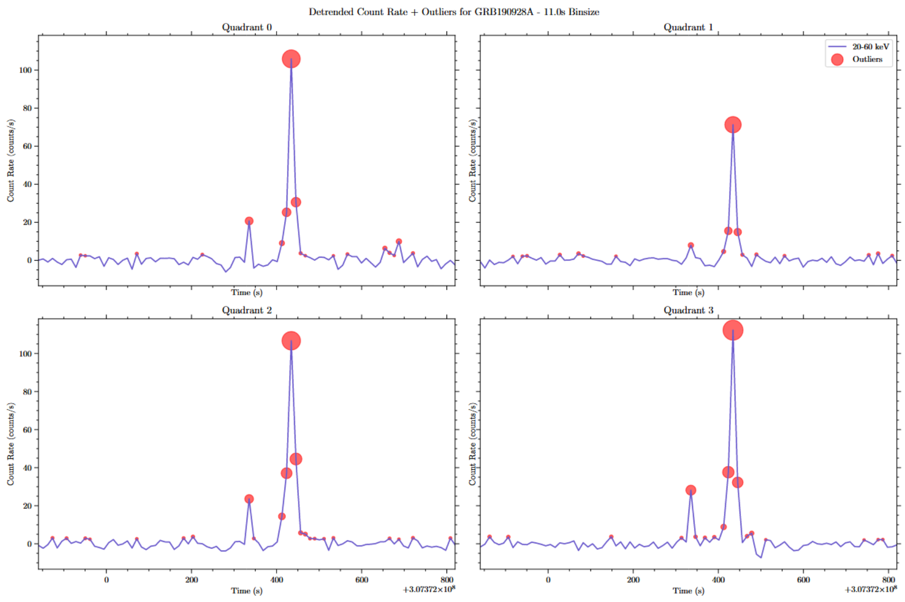
Maximum SNR at $11s$. Plotting lightcurves accordingly:



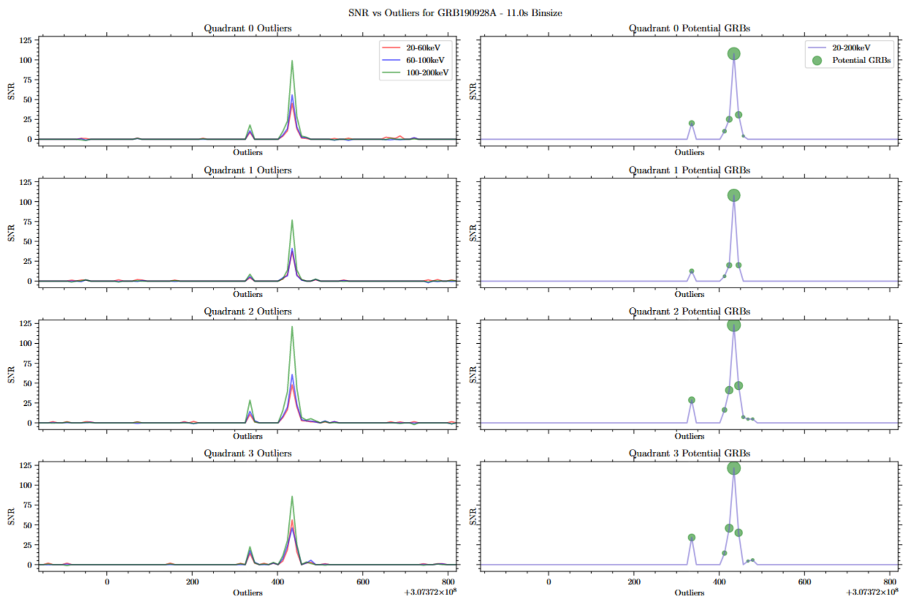
Detrending the lightcurves using quadratic fit:



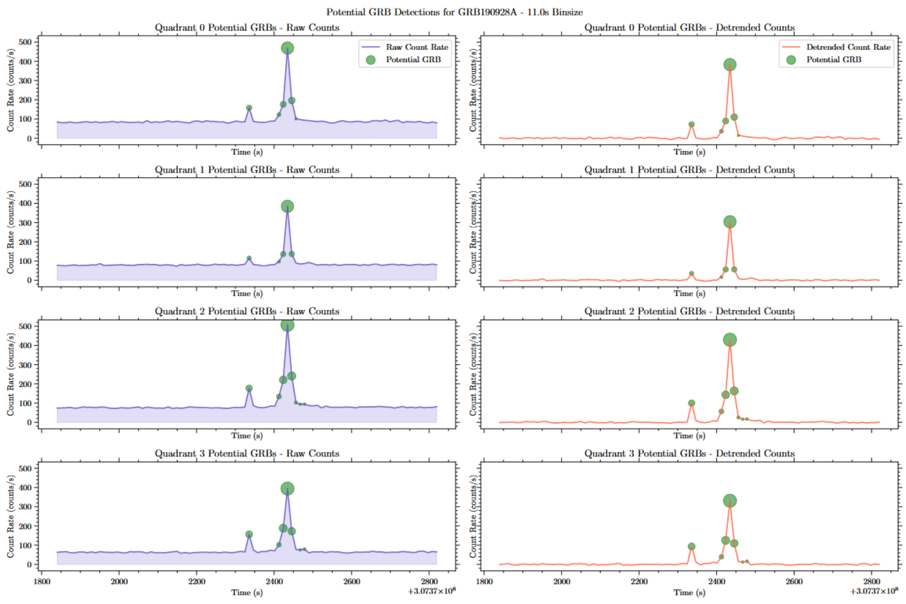
Obtaining first-level outliers by using 1σ clipping on $20-60keV$ energy band to get outlier indices (time stamps):



Using the outlier indices to find SNRs across energy bands and finding second-level outliers by performing 3σ clipping in $60-100kev$ and $100-200keV$ bands:

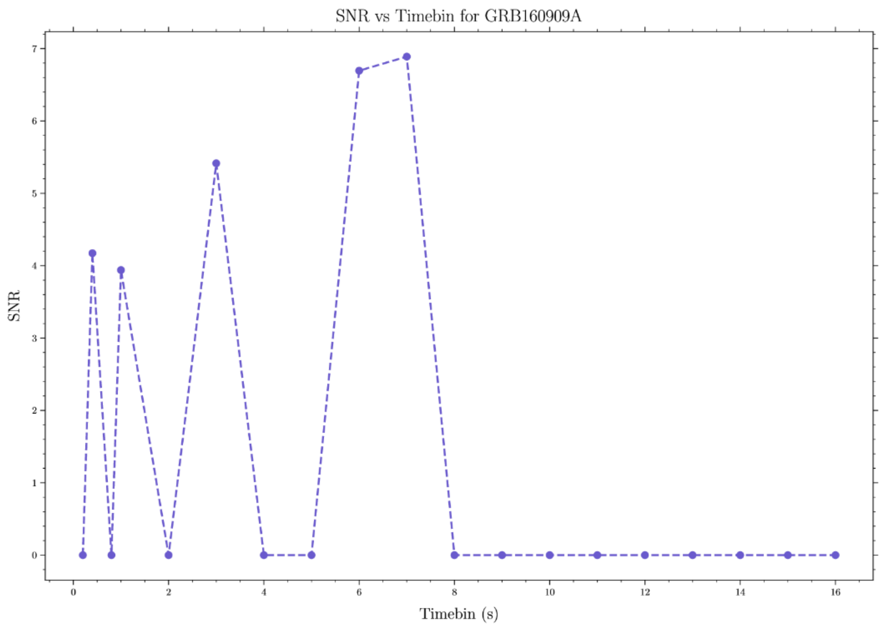


Cross-matching second-level outliers across quadrants to identify potential GRBs:

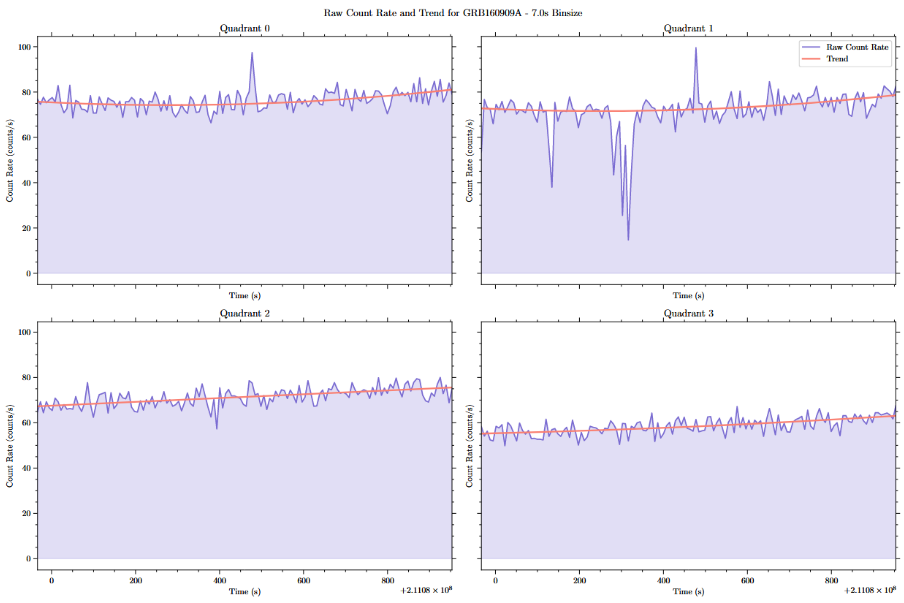


GRB 160909A

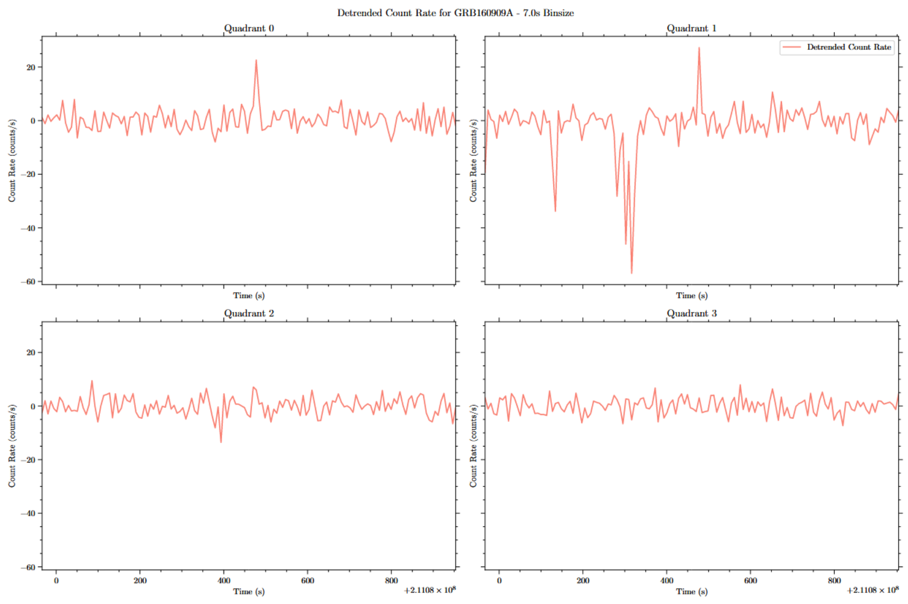
SNR versus time bins sizes:



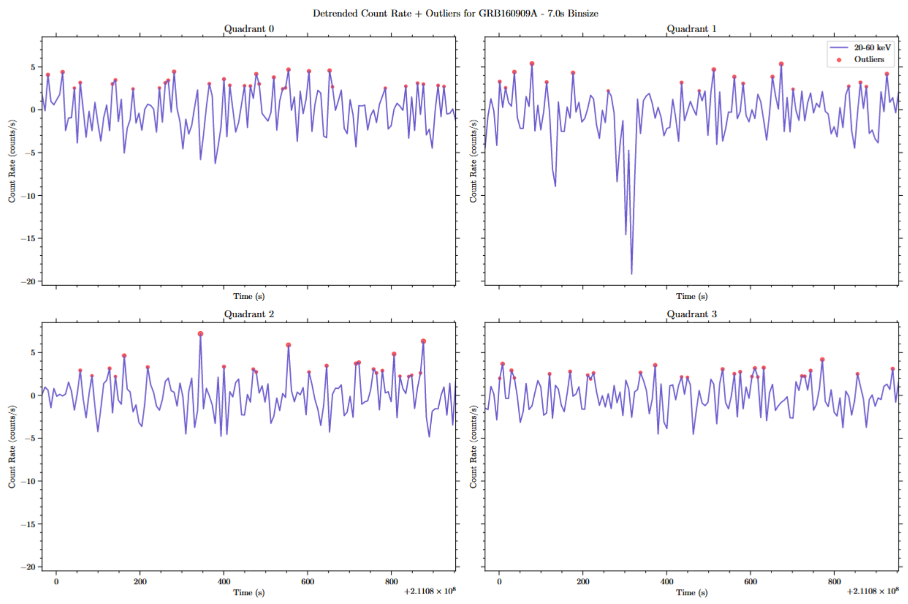
Maximum SNR at $7s$. Plotting lightcurves accordingly:



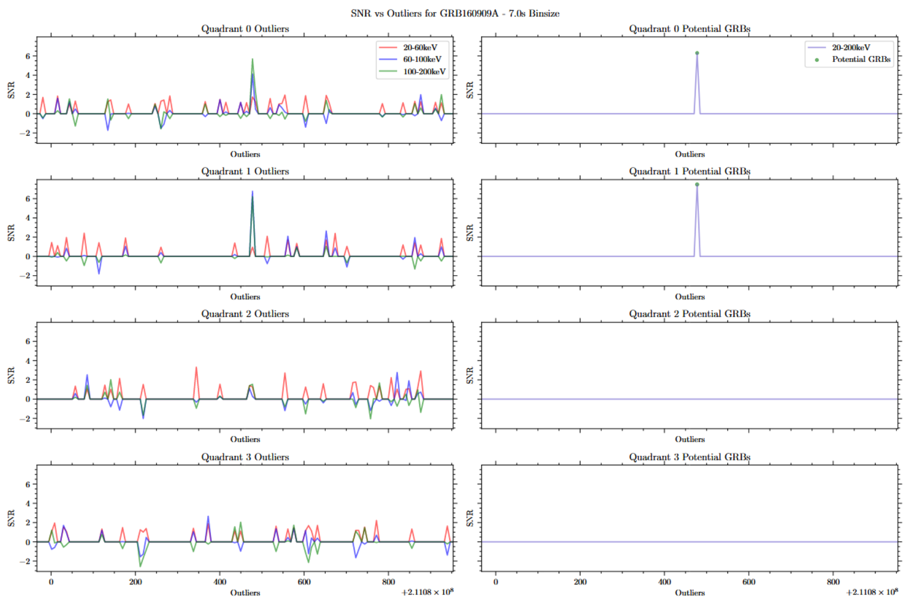
Detrending the lightcurves using quadratic fit:



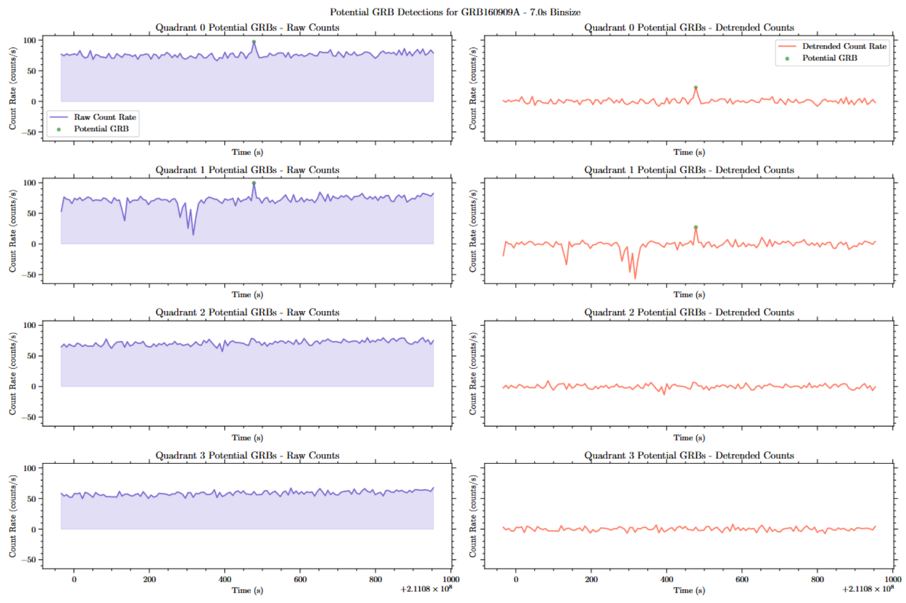
Obtaining first-level outliers by using 1σ clipping on $20-60keV$ energy band to get outlier indices (time stamps):



Using the outlier indices to find SNRs across energy bands and finding second-level outliers by performing 3σ clipping in $60-100kev$ and $100-200keV$ bands:

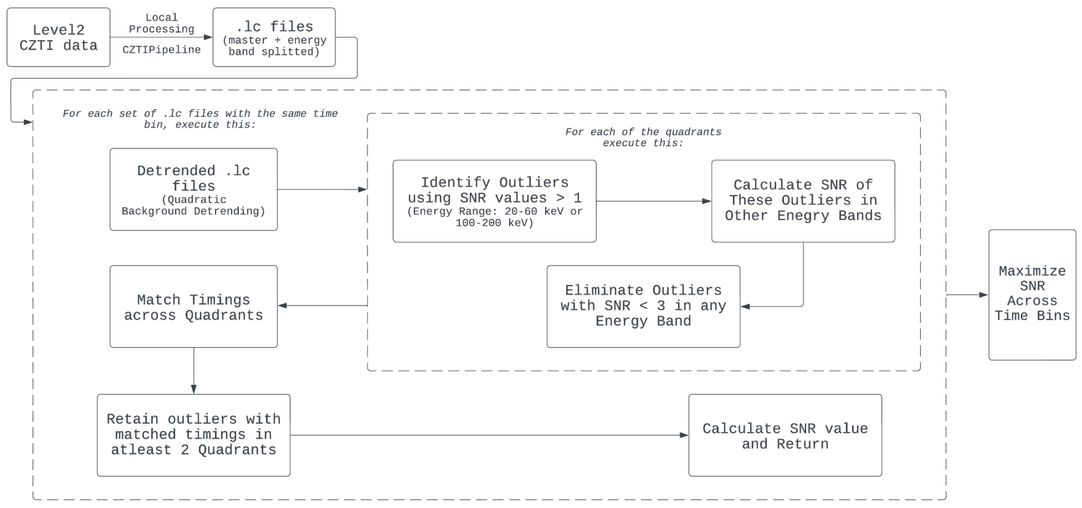


Cross-matching second-level outliers across quadrants to identify potential GRBs:



2.3 Overall Algorithm

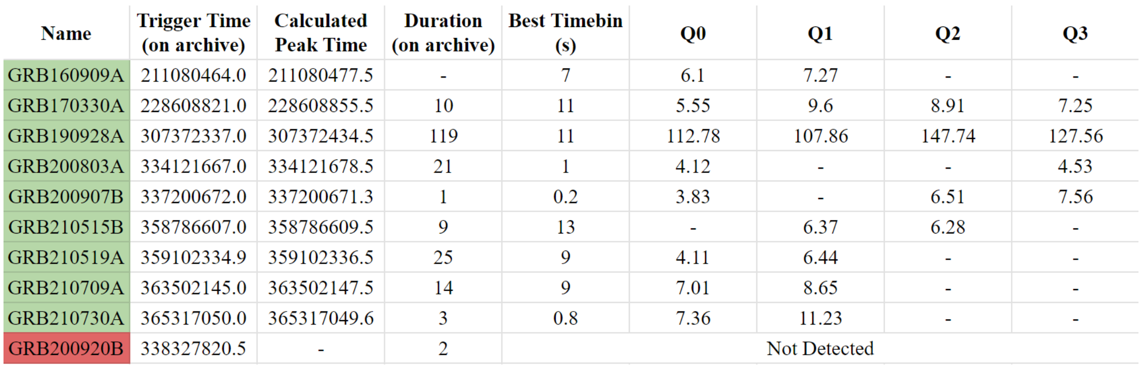
The flowchart now looks like this:


Level2 CZTI data
Local Processing
CZTIPipeline
.lc files (master + energy band splitted)
For each set of .lc files with the same time bin, execute this:
Detrended .lc files (Quadratic Background Detrending)
For each of the quadrants execute this:
Identify Outliers using SNR values > 1 (Energy Range: 20-60 keV or 100-200 keV)
Calculate SNR of These Outliers in Other Enegry Bands
Eliminate Outliers with SNR < 3 in any Energy Band
Match Timings across Quadrants
Retain outliers with matched timings in atleast 2 Quadrants
Calculate SNR value and Return
Maximize SNR Across Time Bins

The files and directories have the below structure:

```
Parent Directory
              ├──► bc.evt
              ├──► bc_livetime.fits
              ├──► level2.mkf
              ├──► mkfThresholds.txt
              ├──► output.pdf
              ├──► 0.2
              │     ├──► 20-60 keV .lc file
              │     ├──► 60-100 keV .lc file
              │     ├──► 100-200 keV .lc file
              │     └──► Master .lc file (20-200 keV)
              ├──► 0.4
              │     ├──► 20-60 keV .lc file
              │     ├──► 60-100 keV .lc file
              │     ├──► 100-200 keV .lc file
              │     └──► Master .lc file (20-200 keV)
              │        .
              └──►     .
                 .     .
                 .
                 .
```

By building a script to perform the above algorithm, we can get the results for multiple lightcurves. Tabulating the results:



Name	Trigger Time (on archive)	Calculated Peak Time	Duration (on archive)	Best Timebin (s)	Q0	Q1	Q2	Q3
GRB160909A	211080464.0	211080477.5	-	7	6.1	7.27	-	-
GRB170330A	228608821.0	228608855.5	10	11	5.55	9.6	8.91	7.25
GRB190928A	307372337.0	307372434.5	119	11	112.78	107.86	147.74	127.56
GRB200803A	334121667.0	334121678.5	21	1	4.12	-	-	4.53
GRB200907B	337200672.0	337200671.3	1	0.2	3.83	-	6.51	7.56
GRB210515B	358786607.0	358786609.5	9	13	-	6.37	6.28	-
GRB210519A	359102334.9	359102336.5	25	9	4.11	6.44	-	-
GRB210709A	363502145.0	363502147.5	14	9	7.01	8.65	-	-
GRB210730A	365317050.0	365317049.6	3	0.8	7.36	11.23	-	-
GRB200920B	338327820.5	-	2			Not Detected		

On comparing the timestamps of potential GRBs from AstroSat CZTI GRB Archive and the timestamps detected by the script, we can see that the trigger timestamps are very close, and a GRB is detected according to our algorithm in most of the lightcurves. In the above table, if a potential GRB is detected, the SNR of the GRB in the corresponding quadrant is recorded for the optimal time bin.