A SEARCH FOR SELF-SIMILARITIES IN BATSE GAMMA-RAY BURST EMISSIONS USING AGGLOMERATIVE CLUSTERING

A thesis submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

in

DATA SCIENCE AND ANALYTICS

by

THOMAS CANNON

AUGUST 2020

at

THE GRADUATE SCHOOL OF THE UNIVERSITY OF CHARLESTON, SOUTH CAROLINA AT THE COLLEGE OF CHARLESTON

Approved by:

Dr. Jon Hakkila, Thesis Advisor

Dr. Ayman Hajja

Dr. Amy Langville

Dr. Michael Larsen

Dr. Godfrey Gibbison,Interim Dean of the Graduate School

ABSTRACT

A SEARCH FOR SELF-SIMILARITIES IN BATSE GAMMA-RAY BURST EMISSIONS USING AGGLOMERATIVE CLUSTERING

A thesis submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

in

DATA SCIENCE AND ANALYTICS

by

THOMAS CANNON

AUGUST 2020

at

THE GRADUATE SCHOOL OF THE UNIVERSITY OF CHARLESTON, SOUTH CAROLINA AT THE COLLEGE OF CHARLESTON

We present a new method for categorizing Gamma-Ray Burst (GRB) emission episodes with similar light curves from the Burst and Transient Source Experiment (BATSE) onboard NASA's Compton Gamma-Ray Observatory (CGRO). We compare normalized time-series data from any two respective GRBs' 64ms light curves using several statistical tests. The comparisons are used in the construction of similarity matrices as input for a hierarchical clustering algorithm. With the new application of this data mining tool, we begin to see similar GRB light curves cluster together by emission properties that exist independent of their amplitude and time scales, leading to a unique understanding of GRB physics.

ACKNOWLEDGEMENTS

This thesis is the culmination of my experience in an incredible new program at the College of Charleston. Clearly, it is not possible to express the gratitude owed to everyone involved in my success, so I hope those not mentioned here know that they are appreciated.

The person most responsible for ensuring that I accomplish something in academics and for ensuring the existence of this work is my advisor, Jon Hakkila. Since my first day as an undergrad at the College of Charleston, his patience in teaching me everything from first principals to final presentation has been a positive forming factor in my life. Also assisting in this endeavor were alumni of our undergraduate research group. Stephen Lesage and Eric Hofesmann routinely served as sounding boards for my strangest ideas.

This research and my success in the program would also not have been possible without my employer Tom Blazer, who encouraged me to pursue this education while maintaining my career. I would also like to thank everyone at my company who picked the slack when I needed it.

I'd like to thank Renée McCauley and the rest of the faculty and staff for their work in facilitating the first cohort and the first thesis of this program. My classmates also deserve an acknowledgment for all the extra help, late nights, and odd hours we all contributed towards this program together.

I am grateful for the commitment and advice from my committee members, Amy Langville, Ayman Hajja, and Michael Larsen, who I have also had the privilege of having

each one as an instructor. I admired the love of their profession and enjoyed every course taught. Other past instructors who have had a lasting impact on my work are Joe Carson whose expectation of quality work encouraged me to push the envelope, and Jeff Wragg who encouraged me to bend the rules when necessary.

My family has always encouraged me to pursue an education. They may not have always known what I was doing, but their support was unwavering. Most importantly, I would like to thank my wife, Jaime. There was never a time where she forgot how important this was to me. Even at times when I forgot, she would remind me and still remain a constant source of support and encouragement.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Historical Context

In 1963, the Partial Test Ban Treaty prompted the United States to launch the Vela satellites in order to monitor and enforce a ban on nuclear testing. The Vela satellites carried the ability to detect gamma-ray radiation from nuclear explosions originating on earth, and in doing so, accidentally discovered energetic flashes of gamma radiation out in space. (Klebesadel et al. 1973). Over the next few decades, the debate over the origins of this phenomena had still not been settled when in 1991, the Burst and Transient Source Experiment (BATSE) on board the Compton Gamma-Ray Observatory (CGRO) launched. The new data from BATSE showed an isotropic distribution of these bursts occurring anywhere around the sky at a frequency of about one event per day. This distribution indicated that the GRB events originated beyond the Milky Way (Paczynski, 1991; Meegan et al. 1992). Years later, the cosmological origin of these events was confirmed when a redshift was obtained on an event named GRB 970228 (van Paradijs et al. 1997), meaning that GRBs occur outside of our galaxy and far back in time. Models to describe the GRB prompt emissions have been proposed, but over almost half a century later, we still do not have a grasp on the mechanisms responsible for causing the prompt emission.

1.2 GRB Emissions

GRB emissions, while their complexities range, have a defined, non-random structure. The cause of which is still highly debated. Examples of GRB emissions and their diversity can be seen in Figure 1.

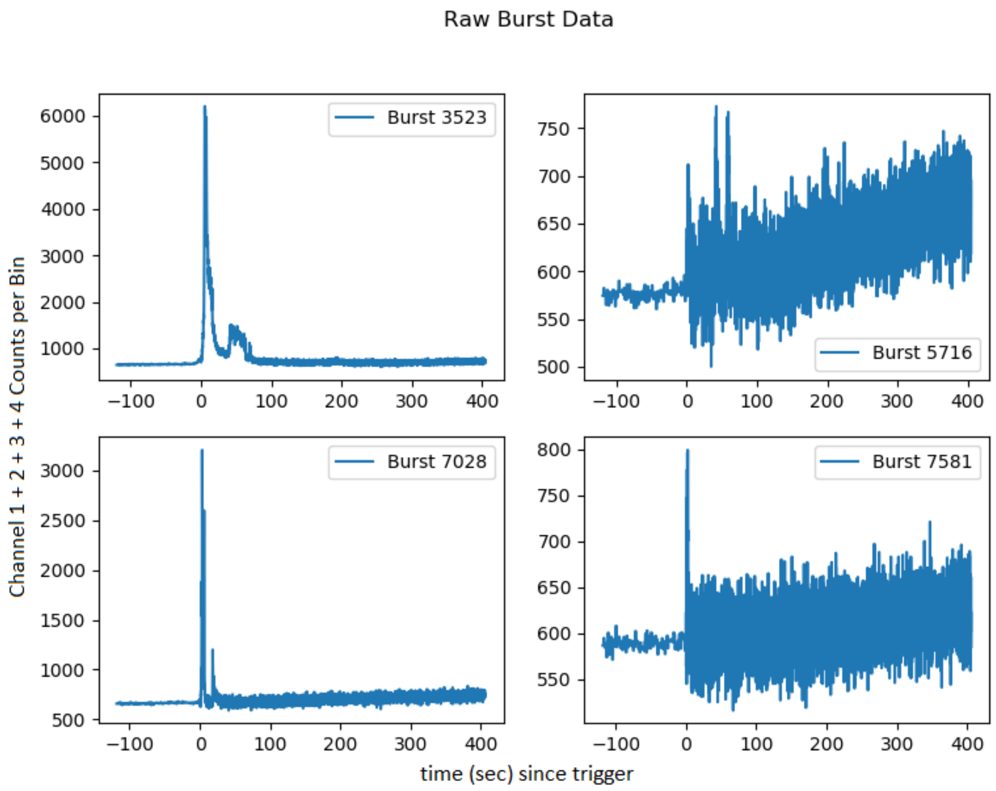

Figure 1 – Examples of Raw Gamma-Ray Burst Data

GRBs are made of pulsed radiation, and in recent years, have been studied thoroughly to give a greater understanding of the physics behind a GRB event. The basic units of a GRB are its pulses (Hakkila & Preece 2011; Hakkila et al. 2015, 2018). The properties of a single pulse have been thoroughly measured (Golenetskii et al. 1983; Liang & Kargatis 1996; Norris et al. 1996; Norris 2002; Ramirez-Ruiz & Fenimore 2000) and can be fitted by a four-parameter empirical model (Norris et al. 2005). The Norris model is a monotonic function used for extracting the shape of a single pulse to several overlapping pulses in a time-series GRB light curve (Hakkila & Cumbee 2009; Hakkila et al. 2008; Norris et al. 2005) as seen in Figure 2.

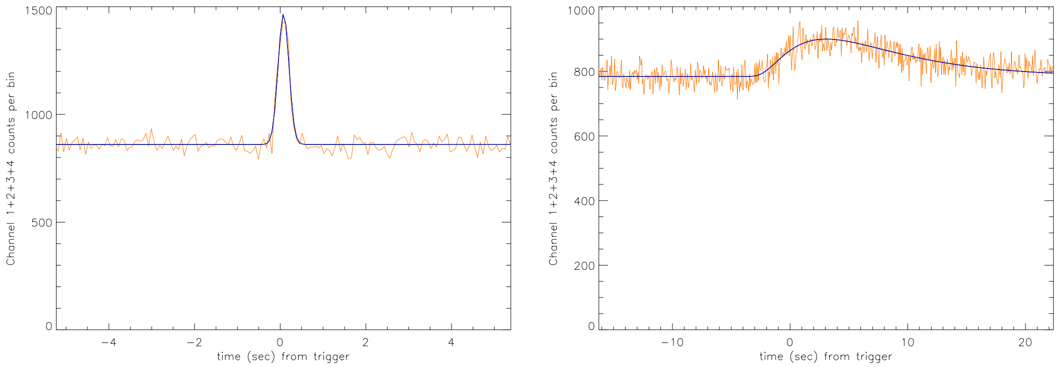

Figure 2 – Two GRBs with Norris Model

We define an emission episode as an increase in the detected gamma-ray flux above the background noise where there is no discernable relationship to any other adjacent increase in flux above the background. Emission Episodes can be made up of a single pulse or what to the eye looks like many pulses. We call emission episodes of what looks like many pulses highly structured. In a structured GRB emission episode it becomes difficult to understand the emission structure and accurately understand the processes of the GRB event (Hakkila & Cumbee 2009). This is because GRB pulses are actually non-monotonic (Hakkila & Preece 2014; Hakkila et al. 2015, 2018).

On top of the monotonic Norris model, GRB pulses exhibit residual fluctuations in phase with the pulse above the background noise. The residuals most commonly appear on top of a pulse as a triple-peaked structure that is approximately centered around the pulse peak. The triple-peaked structure is not always the case. Pulse residuals can propagate more than three peaks and also exist out of phase with the pulse peak. This structure is important to note because it is difficult to understand the evolution of these peaks with respect to signal to noise (S/N), which is defined as

$$\frac{S}{N} = \frac{P_{64} - B}{\sqrt{P_{64}}}$$

where P_{64} is the peak flux of the emission and B is the average background rate (Hakkila et al. 2018). This residual structure more often occurs in bursts with a high S/N, and bursts with lower S/N will typically have this structure washed-out. The washed-out pulses fit the monotonic model well, while bursts with less noise have structure that requires more explanation.

Many GRBs are classified into two different categories: Long and Short (Kouveliotou et al. 1993; Mukherjee et al. 1998; Hakkila et al. 2003), whose bimodal distribution can be seen in Figure 3. These categories were selected primarily based on duration. However, recent work has shown that similar correlative pulse properties not only exist in both Long and Short bursts (Hakkila & Cumbee 2009), but that the Long and Short pulses share common trends of these different property correlations such as duration, lag, peak flux, hardness ratio, asymmetry, and fluence (Hakkila & Preece 2011). This suggests that the Long and Short bursts originate from similar physical processes, which is an important assumption that we are making.

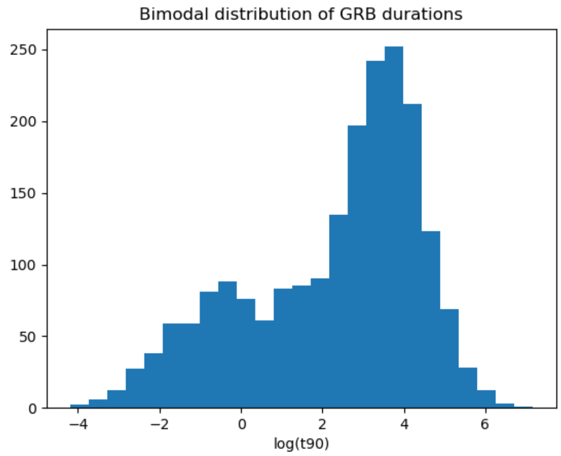

Figure 3 – Histogram of GRB log(T90) Times

1.3 Objectives

Astronomy has a long history of observing complex objects and events with no way to initially build a unified model to objectively explain what intuitively looks to the observer as the same phenomena. One of the most well-known examples is the life cycle of stars – with super giants, Sun-like stars, red dwarfs, neutron stars, etc. We began with an intuition that all of these objects are somehow related and governed by the same physics, but it was not until after centuries of classification attempts and piecemeal understanding of the parts that we developed a more cohesive model to explain all of the avenues and evolution of the stellar lifecycle – which can be generally represented in the Hertzsprung–Russell diagram. Now, with modern computing and an adequate dataset, we have means and motivation to help sort through single GRB events and attempt to explain the relationships between each GRB in order to understand the physics that govern the population of GRBs as a whole.

1.4 BATSE 64ms Data

BATSE (Fishman 1992), as seen in Figure 4 (Mallozzi, R. 2001), was an experiment on board CGRO that launched in April 1991 and operated for over 9 years. It contained 8 detectors on each of the satellite's corners, creating an isotropic view of the gamma-ray sky. When a significant change in the gamma-ray background occurred in the detectors, it would begin recording an observation, counting and binning the number of photons detected from the interaction of gamma-rays with the detector's sodium-iodide based crystals. The data were collected in four different energy channels, ranging from highly energetic X-rays at 20keV to gamma-rays at 1MeV. A GRB can vary in its emission throughout each channel, and in some cases, will not emit above the background enough in one channel to even be noticeable. For the scope of this analysis, we are going to sum the four channels into a single time-series array.

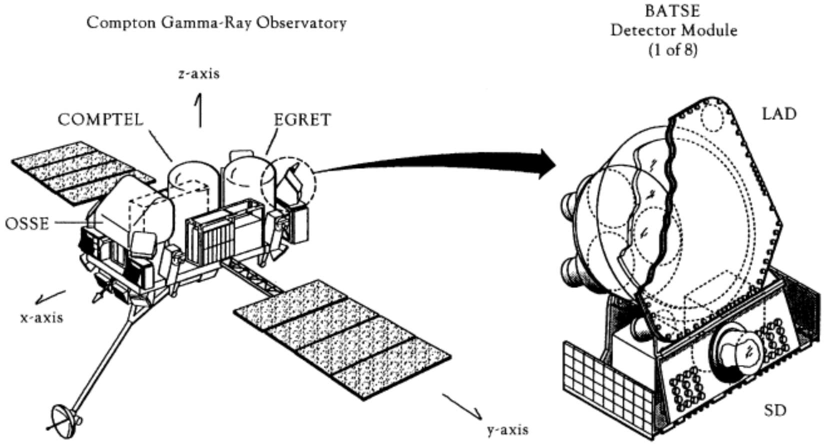

Figure 4 – Diagram of CGRO and BATSE LAD Detectors

The satellite orbited earth in an elliptical orbit, which plummeted the experiment in and out of earth's radiation belts. The radiation belts contributed to the background noise in the detectors, the varying level of which can be seen between any two bursts. There were also other sources of background radiation that muddy the data such as solar flares, X-ray binary systems (*e.g.* Vela X-1), and gamma-ray producing black holes (*e.g.* Cyg X-1). Besides these kinds of background noise sources or a weak detection in one of the four energy channels, S/N can change for other reasons such as an occultation of the source or a failure in one of the detectors or energy bands. This causes some of the data to be unusable for some analysis.

The data used for this study are archived from the life of the experiment. The summed four channel data have a time resolution of both 1024ms and 64ms per burst. The BATSE experiment collected data at 1024ms until a specified trigger criterion was met. Once met, BATSE would then record the remainder of the burst in 64ms time resolution. Because of this switch in time resolution and a trigger criterion changing throughout the experiment, some GRB samples in our working dataset will have both of the time resolutions along their light curves.

The data are freely available to download as ASCII files. The files contain a few lines of meta-data with information such as the number of time-steps of the burst and the total number time-steps since the trigger time followed by four tall columns of count data with each column representing an energy channel and each row representing a time-steps. We sum each column together across the rows in order to produce the combined-four channel data, represented as one column of photon counts per 64ms time-steps.

We will also use the duration table from the BATSE 4B Catalog (Paciesas et al. 1999). This table contains the T90 times, which is defined as the time in which the middle 90% of the photon counts in the burst is observed, where the start begins after the first 5% of the photon counts are observed. We use the time-frames from 90% photon counts window to help put boundaries on the emission episodes for use in preprocessing and normalization.

1.5 Summary

As mentioned above, since GRB emission episodes – which are comprised of pulses – have correlated properties to their duration, fluence, and spectral properties, we have an argument supporting a normalized comparison of the time-series emission data of every emission to every other emission, with the intent of uncovering classifications of bursts that are clustered to one another. If properties were not correlated with time, we could not assume that the physical mechanism responsible for creating features in the time-series data could do so consistently at different time scales. Despite the data being normalized, there are still biases we carry over into the analysis from the raw data. These biases are mitigated through the steps of the clustering process. The three main steps are data preprocessing, building an adequate similarity matrix, and choosing the proper clustering routine. We attempt several different methods between preprocessing and building matrices, leading to multiple pipelines to draw results from. Building similarity matrices from time-series data of different lengths is an area of active research where novel ideas are being tested. Therefore, the definitions of several different ways to build a similarity matrix will be given special attention in section 2. In section 3, we define agglomerative clustering. Section 4 describes how the steps of preprocessing the data and the application of the defined methods for producing the similarity matrices and clustering. Section 5 discusses the biases, strengths, and weaknesses for select permutations of the pipelines from preprocessing to cluster results. It also discusses the results themselves and what it means for GRB physics.

2. SIMILARITY MEASURES FOR TIME SERIES DATA

Binned time-series data are a sequence of real numbers representing the total counts of an event in a given increment of time. We consider a similarity measure as a resemblance value that is calculated between any two vectors (*i.e.* the emission episode time series data) and exists outside the influence of any other vectors. A good survey for similarity measures of time-series data was written by (Liao 2005), and an application and comments on some of these techniques described by (Iglesias 2013).

In order to construct a similarity matrix, we need to generate a value of resemblance between a pair of vectors, creating an upper triangular matrix of values that are meant to represent how similar any two emission episodes are to each other. Given any two time-series sets of GRB data $\vec{x}$ and $\vec{y}$ of equal length n, we present the following methods of calculating a similarity value.

2.1 Euclidean Distance

Euclidean distance has been used in time-series matching and similarity distances for years (Faloutsos et al. 1994). For our purposes, we need to assess the Euclidean distance measure between the differences in the $\vec{x}$ and $\vec{y}$ vectors to create our Euclidean similarity measure d_E. Thus, it is represented by

$$d_E = \sqrt{\sum_i |\vec{x}_i - \vec{y}_i|^2}$$

It is important to note that this measure will only work on vectors of equal length, which can be solved by resampling the binned GRB data. Euclidean distance is also insensitive to time dependent features of a vector. As in, a novel structure in one emission episode can also appear in a different emission episode, but if these two structures are out of phase, the Euclidean distance will not be able to see it. It is blind to feature correlation unless the two features are in phase. Furthermore, while this metric is a good representation of how similar any two vectors are, it's output is not normalized, meaning that unless the values in the vectors are all on the same scale and the vectors themselves are all of a similar length, then the similarity values between each pair – even if the pairs are normalized to each other in time and scale – are not comparable. In other words, the Euclidean distance values from larger dimensional vectors have the potential to be much greater than the values from smaller dimensional vectors.

2.2 Zero-Normalized Cross-Correlation

As stated above, Euclidean distance is blind to the correlation between features due to prominent features being out of phase. One potential way to mitigate that would be to line up two GRB vectors on their most prominent features using a standard cross correlation. While this method works well describing the correlation between any two vectors, it has the same problem as the Euclidean distance measure where, when working with a population of similarity measures between many vectors, the measures are not on a standard scale to make them comparable. A Zero-Normalized Cross-Correlation (ZNCC) (Lewis 1994; Yoo 2009) does not have this problem. ZNCC is widely used in image processing and is used to normalize and measure the similarities between two images of different exposures. We can retool this method to work for one-dimensional vectors as well. ZNCC is given by

$$ZNCC = \frac{1}{n}\sum_{i}\frac{1}{\sigma_x\sigma_y}(\vec{x}_i - \bar{x})(\vec{y}_i - \bar{y})$$

Assuming vectors of equal length, we select the max value from the ZNCC function created by the continuum from stepwise sliding the two vectors beside each other and calculating their correlations. This max is used as the distance measure.

ZNCC attempts to fix the problem of blindness to feature correlation that the Euclidean method has as well as the problem of normalization of similarity measures between multiple ZNCC values. While the normalization is fixed, the shifting around of vectors in search of the max cross-correlation could potentially yield chaotic results.

2.3 Dynamic Time Warping

Dynamic Time Warping (DTW) (Berndt & Clifford 1994) has been used to generate similarity metrics (Keogh 2002), which have been used in clustering and classification (Łuczak M 2016). It does not need to be given vectors of the same length and is created to spot feature correlation. It is widely used in voice recognition to help machines match voice to words where two individuals are speaking the same phrase at different cadences. It finds the best alignment between two sets of the peaks and valleys of speech data to determine their similarity.

DTW allows a non-linear mapping of two vectors by minimizing the distance between them for vectors of lengths that are the same or different. Normally, the two vectors would not have to be of equal length, where $\vec{x} = x_1, \dots, x_i, \dots, x_n$ and $\vec{y} = y_1, \dots, y_j, \dots, y_m$. However, since we will be resampling our vectors to the same length, we can give the vectors the same indices, where $\vec{x} = x_1, \dots, x_i, \dots, x_n$ and $\vec{y} = y_1, \dots, y_j, \dots, y_n$. DTW builds an n-by-n cost matrix C that contains the distances between two points x_i and y_j. Then a warping path $W = w_1, w_2, \dots, w_K$ is formed by the set of matrix components, where $\max(m, n) \leq K < m + n - 1$. In addition, the warping path W should satisfy three local constraints:

1) Endpoint constraint: $w_1 = C(1,1)$ and $w_K = C(i, i)$
2) Monotonicity constraint: if $w_k = C(a, b)$ and $w_{k-1} = C(a', b')$, then $a \geq a'$ and $b \geq b'$;
3) Continuity constraint: if $w_k = C(a, b)$ and $w_{k-1} = C(a', b')$, then $a \leq a' + 1$ and $b \leq b' + 1$.

See Figure 5 for an illustration of the distance geometry. There are many warping paths that satisfy these conditions. The warping path that minimizes the warping cost is considered the DTW distance and is what we use as our similarity measure. Here we minimize over the cost matrix to find the warping path, which is used to tell us which series of points between the two vectors should be compared.

$$d_N = min\left(\sqrt{\sum_{k=1}^{K} w_k}\right)$$

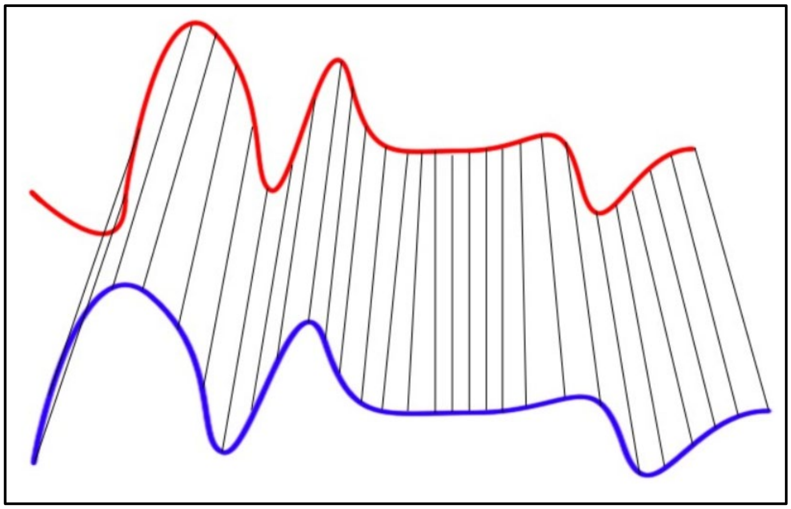

Figure 5 – Illustration of Distance Geometry in DTW

A potential problem with DTW is that while it adjusts for a phase shift between two time-series, it can perhaps be too liberal and allows for a variable phase difference in features. As in, the duration of and between features could potentially be less relevant than the order in which the features appear in. While DTW does take a few liberties in the durations between features between any two vectors, there is still a punishment in the fact that the Euclidean distance within DTW will be greater between two vectors whose feature durations cannot easily scale to one another over two vectors whose feature durations can scale to fit one another well.

2.4 Normalized Manhattan

One last similarity test that is used as a simple calculation for comparing dataset is the Manhattan distance, also known as the taxi distance. It is like the Euclidean distance, but instead of measuring the distance by direct line of sight, it uses the total distance between the two points as if one were driving a taxi through city blocks. Under normal circumstances, Euclidean distance is the preferred method over Manhattan when building similarities for clustering. However, we have a unique case where the scale of our vectors will be normalized between 0 and 1. This will allow us to divide by the length of the vectors in order to effectively take an average Manhattan distance between any two vectors that we can guarantee that our similarity measure will fall between 0 to 1. This means that we can confidently compare the similarity distances calculated from any pairs of vectors. The metric is defined by the Manhattan distance divided by n:

$$d_M = \frac{1}{n}\sum_i |\vec{x}_i - \vec{y}_i|$$

3. AGGLOMERATIVE HIERARCHICAL CLUSTERING

An Agglomerative Hierarchical Clustering algorithm works from the ‘bottom up”; meaning it begins with every element in its own cluster. As described by (Jain, et. al. 1999), and modified to represent GRB light curves, the algorithm proceeds as follows:

1. With an upper triangular matrix, where each entry contains the similarity distances between every unique pair of GRB emission episodes, treat each entry as its own cluster.
2. By the values in the matrix, the two GRBs considered the most similar out of the entire matrix are then merged into one cluster. The matrix is then updated to reflect the merger by considering new position values that represent the combined cluster.
3. If every GRB emission episode is within the same cluster, then stop. Otherwise, return to step 2.

There are several ways to calculate a new cluster’s position. Most commonly used are the single, complete, and average methods. We use the average linkage method, which uses the average center of a cluster of samples as the location of the cluster. We base our decision for which linkage process to use on precedent set in previous GRB analysis (Hakkila & Preece 2014). A set of GRB pulses were normalized based on their Norris function fit. These normalized pulses were then averaged into a single empirical GRB pulse. The motivation in doing so was to look for underlying residual structure that is common among all GRB pulses with the idea that if no underlying residual structure exists, then by averaging pulses together, one would not see any significant structure. The combination of these pulses showed residuals that could not be explained by Poisson noise. This discovery lead to a residual model that supplements the Norris model, improving the chi-squared fit. Since the method of combining these pulses resulted in newly discovered structure, we use the average linkage method when computing clusters' centroids. The intent is to produce as many stand-alone branches of GRB emission in the dendrogram that can considered to have unique characteristics as possible. In a sense, if one was to align and average all the GRB emissions episodes in a given branch, one would see a unique structure not apparent in any other averaged branch.

There are several packages in programming languages that will not only calculate these similarity distances but will also take the similarity matrix inputs and return clustering results. In our case, we use the Python SciPy library (Virtanen, P. et al. 2020), supplemented by the NumPy library (Stéfan van der Walt, S. et al. 2011) to conduct the preprocessing and clustering, written into several scripts for each step.

Agglomerative Hierarchical Clustering is used in this analysis because the user does not need to know the number of clusters before-hand. Clustering techniques like K-means require the user to tell how many clusters it should find before it begins. This is not so with agglomerative, where the final cluster count is all dependent on where the cutoff is set at the end of the analysis.

4. METHODS

4.1 Preprocessing

Before any similarity matrices could be created, we needed to sort through every GRB collected by BATSE and eliminate bursts with inadequate data. The BATSE 64ms ASCII data are a raw dataset and are rife with irregularities that caused some data to be unusable for our purposes and needed to be filtered out. There are three large filters that every burst passed through.

The first filter ensured that every burst had a proper T90 time in the duration table. Because we used the T90 times to help normalize our data, we needed to ensure that our dataset began with a union of the bursts available in the T90 table and the bursts with proper ASCII files. There was only one burst found with a corrupt ASCII file, but several missing from the duration table. After this union, the total number of useable files dropped from 2139 to 2041. The duration table was pulled from the BATSE website (Paciesas et al. 1996) and condensed to a CSV file for ease of use in the file, *duration_table.csv.*

The second filter was a manual check to see if every burst contained a single emission episode. We did this based on our assumption that the mechanisms that produce the GRB prompt emission create a common structure in each emission. Even though there are relationships between different emission within the same burst, we do not have the ability to programmatically analyze multiple emissions because the T90 times were created on the entire burst, thus making it not possible to carve a window around a single emission. There was also no current database available with counts of the number of emissions, so we plotted and checked every burst by hand to see if there were multiple instances of an event rising above the background noise. While we searched for bursts of single emissions, we also paid attention to the quality of data and consulted the comments table (Paciesas et al. 1996) when necessary. There were several instances where the data would drop to zero in any given energy channel during the middle of an emission, a solar flare interrupted the emission, an occultation of the source occurred, the experiment failed, or another background source of gamma radiation would interrupt the emission. All of these reasons caused the data to not be useful for our analysis and would be discarded. After this filter, the remaining set of useable data files numbered 1902. A complete list of bursts that passed this filter can be found in the file *burst_info.csv.*

The third filter was to eliminate the categorical short bursts. In order to compare these time series data, the vectors needed to be of the same length. However, in almost every comparison, the two vectors were of unequal length and needed to be resampled to be comparable. Resampling an emission episode with fewer number of time-steps up to a larger amount would not work, because we would be creating data that did not necessarily exist. We therefore would resample emissions of a larger number of time-steps down to match the emission with a smaller number of time-steps. In order to prevent the resampling of hundreds of bursts down to a number of time-steps where the structure of the data gets washed out by resampling to a much smaller number of time-steps, we decided to truncate our dataset to only include emissions with a duration of 2 seconds or longer. The 2 seconds is generally accepted to be the limit where short bursts categorically begin, and it is familiar to the GRB community.

We then summed the four energy channels into one. Individually, the data in each of the four channels are of lower resolution than their sum. Not only does background vary in each of the channels, but some spectrally soft bursts will have no emission in higher energy channels at all. Since a burst had the chance to be missing a leg or two of data due to spectral softness, we summed all four channels for a composite view of each emission so that we would not have to discard more emissions and make our sample set smaller.

The last step in preprocessing the data was to calculate the background around each emission and subtract it. Since BATSE experienced continually changing background noise, almost every GRB had a sloped or changing background that needed to be corrected to keep the data in order. A simple linear regression on the background works fine in an ideal situation. However, the background often changed fast enough to be noticeable within a single burst and represents itself as non-linear. Because the background is caused by several different sources, there was no function that could robustly fit each one. We therefore resulted to using a simple linear model on the background noise located on the flanks of each emission.

Using the T90 times, we found the middle 90% of the emission. A buffer was added to either side of the middle 90% to mask the entire emission. The buffer was added to ensure that the entire emission was captured within the window. A margin was then continued out from either side where the buffer left off. The two margins are slices on the outside edge of the buffered window that contain only background noise. The background noise on either side of the buffered emission window were used to calculate the background slope.

The background is quickly calculated with *scipy.stats.linregress*, which returned the coefficients needed to level and zero the background. A complete list of background calculations can be found in the file *background_table.csv.* In some cases, it was not possible to include the result of the background in the similarity matrix calculations due to bad results from the linear regression or an inadequate amount of margin. The final filter plus the emissions with background that were not calculable or included in the matrix calculations brought the total number of useable GRB emissions to 1310.

4.2 Constructing the Matrices

We began creating the similarity matrices by using the data from emissions that have been zeroed and leveled. From here, the pipeline had two paths it could continue down. We needed to compare only the emission episode, so the adjacent background was trimmed away. However, trimming by the T90 times truncated the emission since the T90 times only represent the middle 90%. If we assume that the middle 90% of the emission is an accurate representation of its whole, then using the T90 times will hold up. In other words, if we can show that between any two emissions with nearly identical T90 windows that the remaining emission outside of the T90 windows is also consistent between the two, then the use of T90 works. While we could not test this concept at the time, we used this concept of changing time windows to tell how robust each of the similarity values was towards the introduction of new background noise and potential phase shifts between the emission episodes, as discussed later.

The second path that the pipeline took was to add a buffer to the T90 times to ensure that the entire emission was visible. Since we assumed that all GRBs originate from similar physical processes regardless of their durations, we constructed the buffer as a function of the T90 durations. To create the buffer, we simply extended the window on the start and end of the emission by a constant multiple of the T90 time of the burst. The buffer option and the strict T90 option gave us two different windows to build similarity matrices from to later test for robustness.

Now that the raw data were zeroed, flattened, and had their windows defined, we ensured that the vectors are the same length and of comparable scale. We determined the size of each emission window and resample the larger one down to the size of the smaller. This was done with the *scipy.signal.resample* function. This function uses a technique that takes the discrete values of the longer vector and downsamples them to the size of the smaller vector by de-discretizing the time-steps through interpolation. The scale of the vectors was then normalized from 0 to 1 by dividing by the max of each vector. This gave us two vectors of equal length with values ranging from 0 to 1 for any pair of emissions no matter how long or energetic they originally were.

To calculate the distance between each set of emissions, we used the four similarity calculation methods described above. These four methods plus the two options of using the T90 buffer or not gave us 8 possible matrices to choose from. For each method, every emission was compared to every other emission and recorded in a flattened upper triangular matrix. Each method was run on the available 1310 GRBs files. The matrices for each and a list of the bursts contained in them were all saved in pickled python files for easy consumption by the following cluster step. A pickled python file is a file made of serialized data structures native to the python language.

4.3 Clustering

We continued with the SciPy python package and used the *scipy.cluster.hierarchy.linkage* function on the data from the pickled python files. As described above, we used the average linkage method. We also used the *optimal_ordering* flag, which for a few more seconds of processing time, organized the data so that any dendrograms produced from it exhibited a more intuitive tree structure. The *optimal_ordering* flag did not change any of the clustering process itself, just the order in which the leaves were visually displayed along the x-axis.

We produced a dendrogram with the *scipy.cluster.hierarchy.dendrogram* function. The two necessary inputs were the main output of the *linkage* function and a list of the bursts that were contained in the original similarity matrices, which were saved in the pickled python files. An example dendrogram for one of the matrices can be seen in Figure 6. The colorIt is immediately evident that the tree is too large to gain any sort of insight about the clusters, so we can zoom in the sub trees as seen in Figure 7.

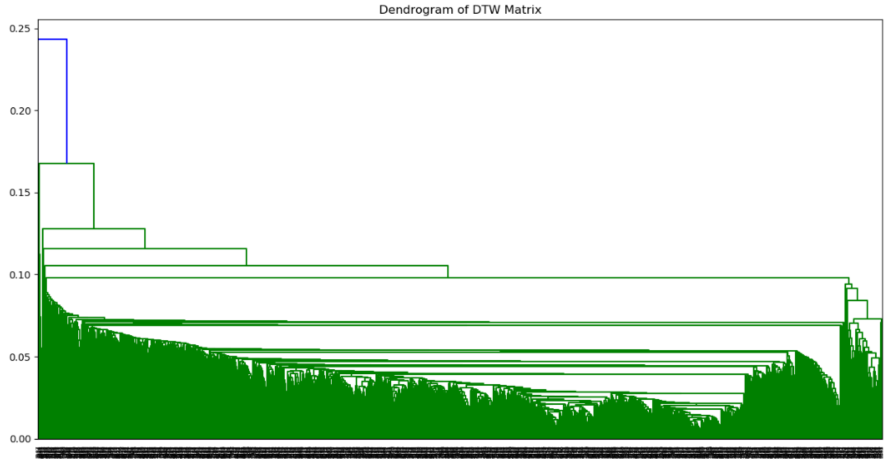

Figure 6 – Full Dendrogram Example

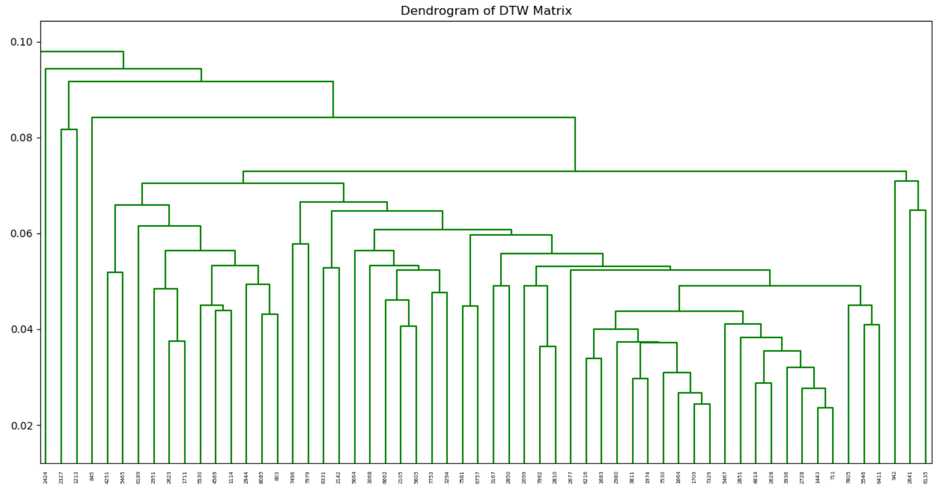

Figure 7 – Partial Dendrogram Example

The colors in the dendrograms throughout this paper are the SciPy package's default attempts to identify and color code clusters. SciPy allows the user to set this color threshold value, which changes the colors based on the vertical dendrogram distance. However, the functionality will not be used in this analysis because we do not yet have the ability to assign as value to the color threshold or evaluate its usefulness.

In order to test how our similarity matrices respond to changes in the T90 times, we use the Pearson correlation. The script that outputs our similarity matrices conveniently does so in a flattened matrix. This created two matrices in a flattened vector form for each similarity calculation. Using Pearson, we found *R* values for each pair of matrices, with the assumption that a lower *R* value between two similarity matrices – where one strictly use the T90 times and the other used the T90 times plus a buffer – would indicate that the given measure was more robust to temporal phase changes, T90 errors, and emission structural shifts.

5. RESULTS

With the clustering completed, we moved the data into a dendrogram for easy visualization. With our large dataset, the dendrogram becomes too large to visualize on a single page. The dendrogram visualizes different emissions as leaves in branches. Each branch represents a cluster of emissions that have similar characteristics. The vertical length of a branch is a measure of how similar an emission or cluster of emissions is to its connecting emissions or cluster of emissions. Sometimes, the tree will display several adjacent emissions in a tight cluster that show evidence for an evolving continuum of GRB properties. There are some emissions that are almost completely unique. These are displayed as branches on the tree whose nodes, common to the rest of the tree, break off very high up. In the algorithm, these emissions would have been selected last as a comparable relative to any other emissions or cluster.

What we discovered in using several types of similarity measures to construct different matrices were two large caveats that ultimately ended up driving our final pick of which similarity measure to use: phase shift and similarity metric normalization. Phase shift is a problem where, because of errors in calculating the window of the emission, any two emissions might not be as aligned as they should. Similarity metric normalization is where the similarity metric values between pairs of emissions are not all normalized on the same scale. This is mostly due to the fact that, because any two emissions are scaled to each other, the length of their vectors are most likely going to be different than the length of the vectors from a different pair. It would be similar to comparing the entire genome of two very different species with a different number and length of chromosomes.

After we created clusters – seen in Figures 6, 8, 9, and 12 – we noticed differences in the vertical distances, called Within Cluster Distance (WCD) by the SciPy documentation, that the leaves connected to an adjacent cluster. The WCD seen on the y-axis can be used to tell how similar the hierarchical clustering found adjacent leaves. A cluster with a high median WCD across all of its leaves would indicate that the metric used found less similarity than a cluster with a low median WCD. For example, in Figure 8 the Euclidean metric cluster shows a higher WCD than in Figure 6, the DTW metric cluster. While the Euclidean dendrogram may look more digestible because of its clear clusters, this is false hope because the overall clustering performed worse with a median WCD of 2.88 compared to the small median WCD of 0.03 for the DTW cluster. Manhattan and ZNCC also have larger median WCDs of 0.11 and 0.32 respectively.

5.1 Euclidean Matrix Cluster

Despite a Euclidean similarity being a commonly used method, in our analysis, the metric does not perform well. The Euclidean distance works on datasets whose vectors are of the same length and each have a comparable scale of data. Raw GRB data have neither. Since the metric does not contain a normalization, it clusters together emissions that fit together well and have large vectors. If either emission has a small initial vector, then the resulting Euclidean distance measure is small because the larger vector is resampled to the size of the smaller. The higher dimensional space that a pair of long vectors exist in naturally gives a larger Euclidean distance over a pair of vectors in a small dimensional space. This is evident during a qualitative inspection of the Euclidean dendrograms. Adding a buffer around the T90 window only exaggerates these effects.

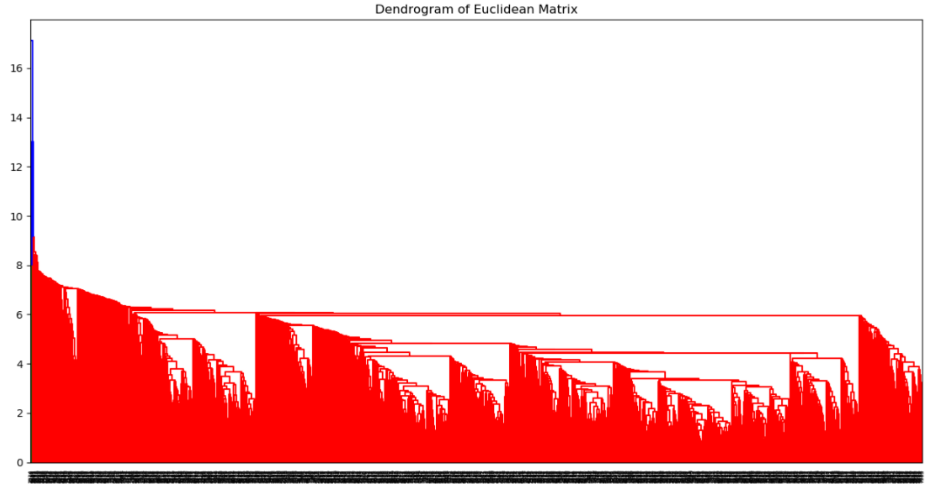

Figure 8 - Euclidean Dendrogram

5.2 ZNCC Matrix Cluster

ZNCC was initially proposed to solve the potential temporal phase errors. It also does not share the same normalization problem as Euclidean. The method has normalization built into it. However, the cross-correlation component of ZNCC allows for overfitting. The expectation was that a small lag adjustment from cross correlation would better align the emissions and produce a quality normalized similarity value. Instead what is immediately evident under a qualitative inspection of the dendrograms was that the emissions' correlations present strong values when lined up on background noise. In other words, the cross-correlation component often settles on its answer off of an accidental alignment on the peaks and valleys of noise from one emission to the peaks and valleys of noise on another. This caused nonsensical results as seen in Figure 9. The gradual slope of the dendrogram means that a single large cluster is constructured fomr the agglomerative “bottom up” process. Clusters containing a just a single or handful of emissions were added one after the other to the main cluster. In a 2-D scatter plot representation, this would look like a single large, loose cluster, giving little valuable information of the differences and similarities in the emissions. Adding or removing a buffer around the T90 window in attempts to add more data and wash out the noise makes no difference.

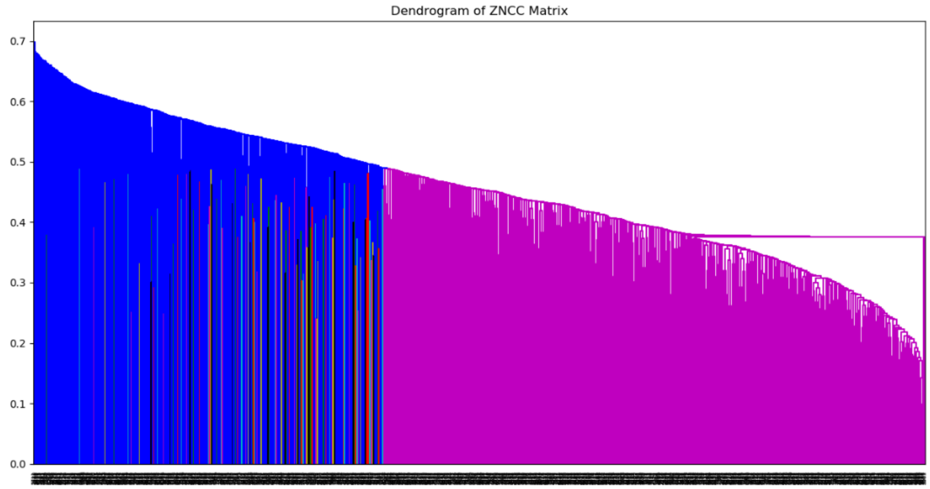

Figure 9 - ZNCC Dendrogram

5.3 Normalized Manhattan Matrix Cluster

After a qualitative inspection of the tree, it is obvious that the Normalized Manhattan matrix works better than The Euclidean distance and ZNCC matrices. It performs well with emission episodes that most resemble the Norris function. We refer to these types of GRBs as canonical single pulses, as can be seen in Figure 10.

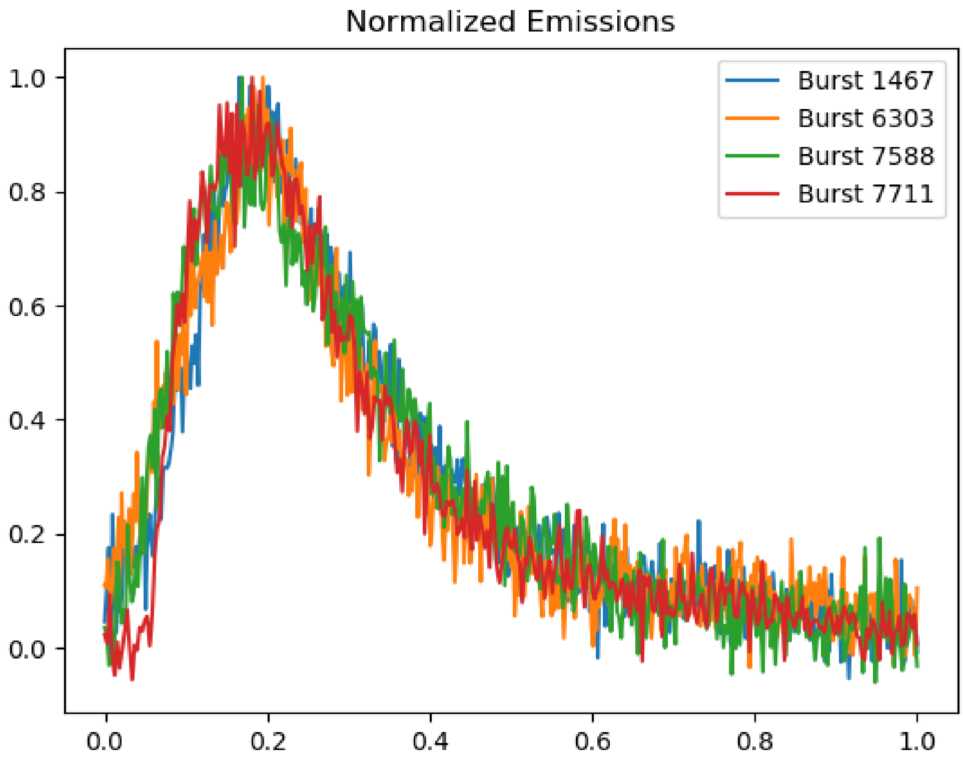

Figure 10 – Canonical Single Pulse GRBs

The reason that Normalized Manhattan distance works so well with these kind of emissions and not others is because of two biases that are introduced. One bias, that also effects the Euclidean method, is the T90 error. Bursts that have especially low S/N will naturally have a higher error in the T90 times, causing the window that we selected based on the T90 times to potentially set up emissions slightly out of phase. Any out of phase shift between two otherwise similar bursts could cause them to produce poor similarity results based on the Euclidean and Manhattan methods. The other bias that Manhattan has is one towards the level of structure in an emission, which is also shared by Euclidean. High levels of structure in GRBs usually present the emissions with high S/N, but a very spiky appearance – see Figure 11. As in, the higher the S/N the more structure we typically see (Hakkila 2020). This structure is often chaotic and unique, which creates many peaks and valleys in the light curve that produce poor similarity values between two emissions.

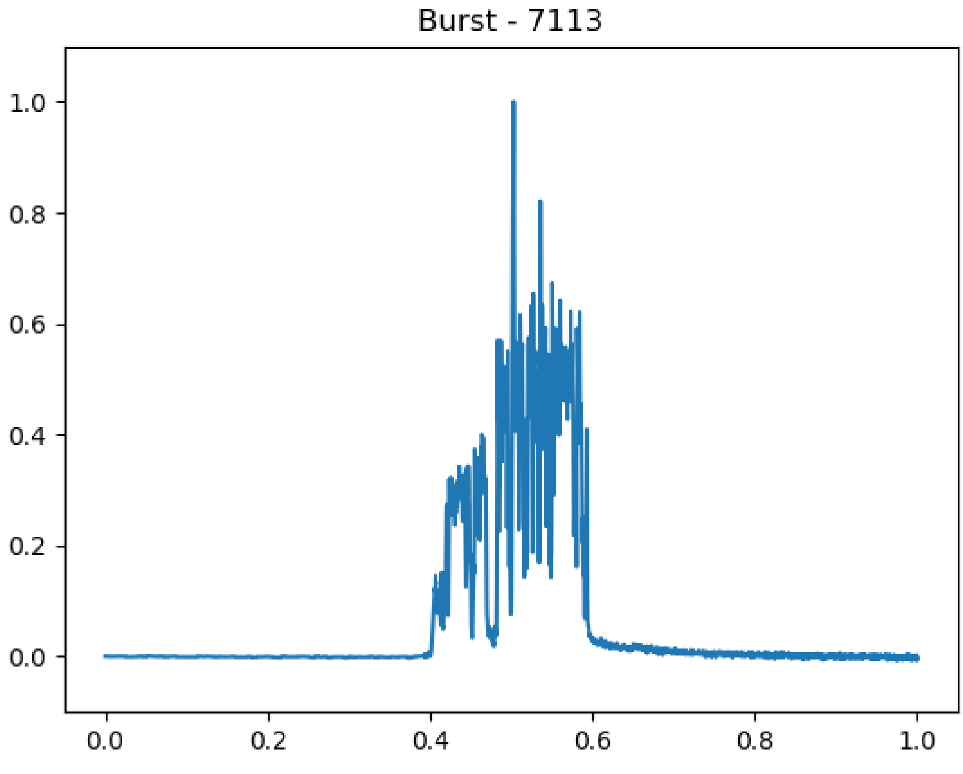

Figure 11 – Highly Structured GRB Emission

Manhattan distance is sensitive to the emissions in Figure 10 because the emissions exist in a medium between the high S/N that produces high structure and the low S/N, where the jagged background causes poor results. The clusters also spill over into one another. As in, an emission that one would think would exist in one cluster exists in the ones over and vice versa. This is probably due to the Manhattan block aspect of the distance measure, which does not give as accurate a representation of distance as a line-of-sight metric like Euclidean.

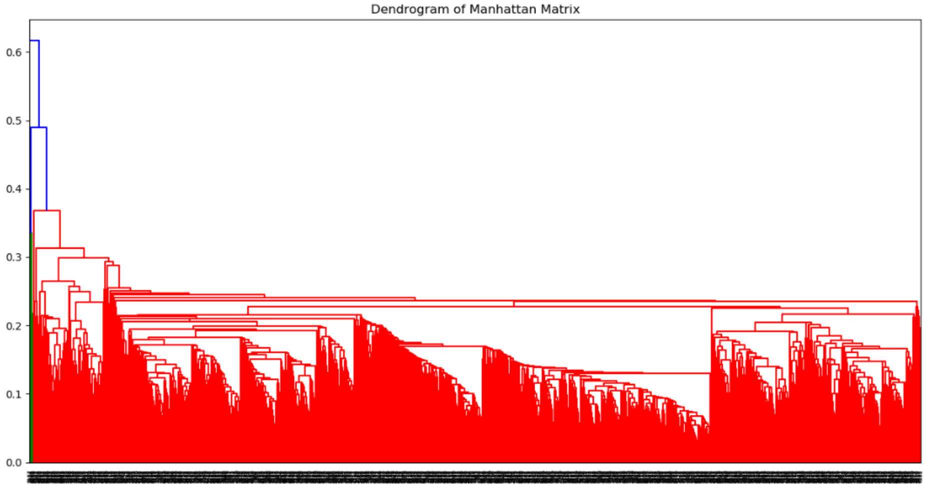

Figure 12 - Manhattan Dendrogram

5.4 DTW Matrix Cluster

DTW is qualitatively the best at clustering similar GRB emissions. A matrix clustered from DTW measures is more sensitive to emissions of simpler structure. When DTW is comparing two emissions with a large amount of structure, it easily warps the large number of random spikes within one emission onto another, which inflates the DTW value and is overfitted. However, as mentioned above, bursts with incredibly high amounts of structure – Figure 11 – are rare and increasingly unique; so, its confusion at these extremes is forgivable with a smaller sample set from which to build clusters. For bursts of a simpler structure, it visually works well. Figures 13 and 14 contain several plots of emissions whose leaves were directly adjacent to one another in the dendrogram. This means that they uniquely share more similarity to each other than any other emission or cluster in the matrix. As seen in these figures DTW finds comparable emissions even when the normalized emissions do not line up perfectly correct based on the T90 windows. In Figure 9 it is evident how the start of the emission in each frame begins at a different time along the x axis, yet still is able to pick out the three-pulsed structure in each emission. The same can be seen in Figure 10, where burst 1443 ends around 0.75 and burst 2728 ends around 0.85 while DTW still has picked out the prominent double peaks in each of the two pulses.

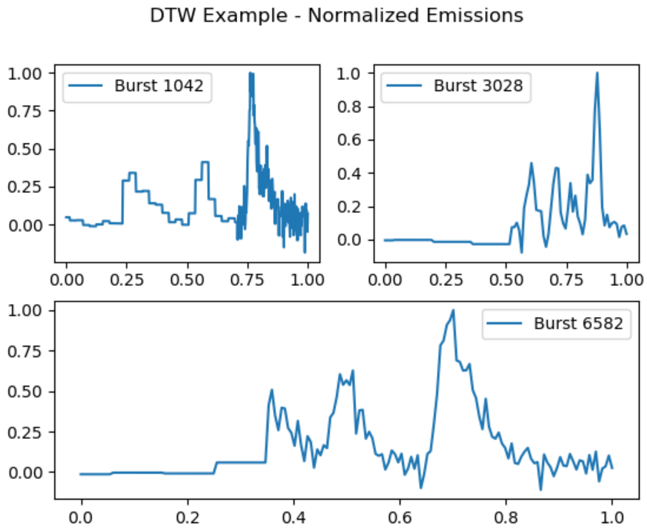

Figure 13 – DTW Adjacent Emissions - 1

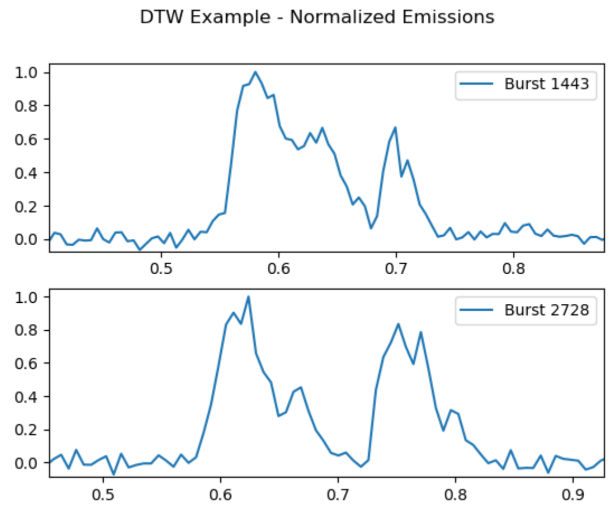

Figure 14 – DTW Adjacent Emissions - 2

Figures 15 and 16 also are leaves of the dendrogram adjacent to one another, but we begin to see the biases in DTW through these examples. In Figure 15, we see four peaks of decreasing amplitude in each emission. From our experience working with these light curves, having an emission with four such peaks is a rare event, so it is encouraging that these emissions were set beside each other.

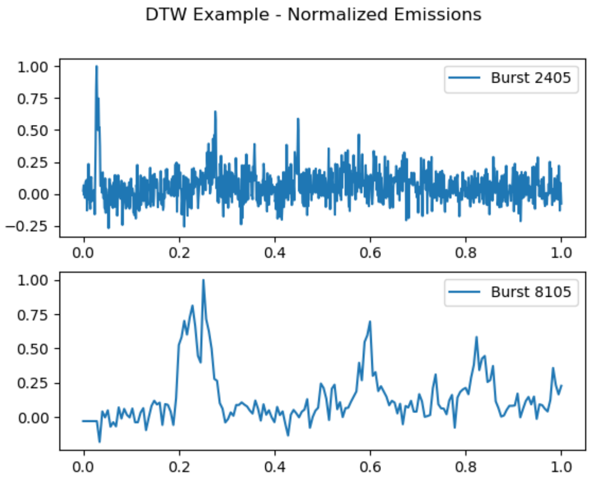

Figure 15 – DTW Adjacent Emissions - 3

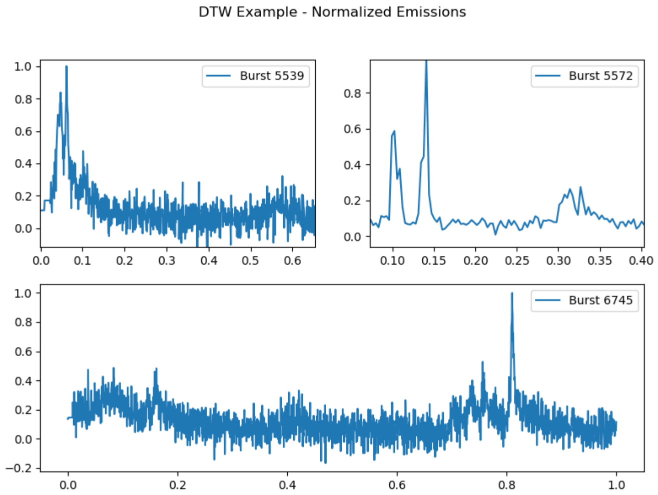

Figure 16 – DTW Adjacent Emissions - 4

Not only are the leaves of the DTW cluster more representative of similar features despite temporal hang ups, but the organization of the DTW matrix was also qualitatively better than the organization of other clusters from other matrices. Examples of its effectiveness can be seen in Figures 17 – 21. Each of these figures is a part of the large dendrogram exhibited in Figure 5. In successive order, the areas in Figures 17 – 21 are randomly chosen from left to right from the bottom of Figure 6. One interesting point to notice is how the S/N increases as we progress from left to right through the dendrogram with the exception of Figure 21, which is located in the last major branch on the far right of the dendrogram. This makes sense when one thinks about it. DTW has a more difficult time with noisier data. If the data between two emissions are noisy, and DTW attempts to create a similarity measure, it will not perform well, and the resulting metric will isolate that point farther from the centroid of a more robust cluster. Hence the visualization where the separation point of the cluster in Figure 17 is higher on the *y* axis than other clusters.

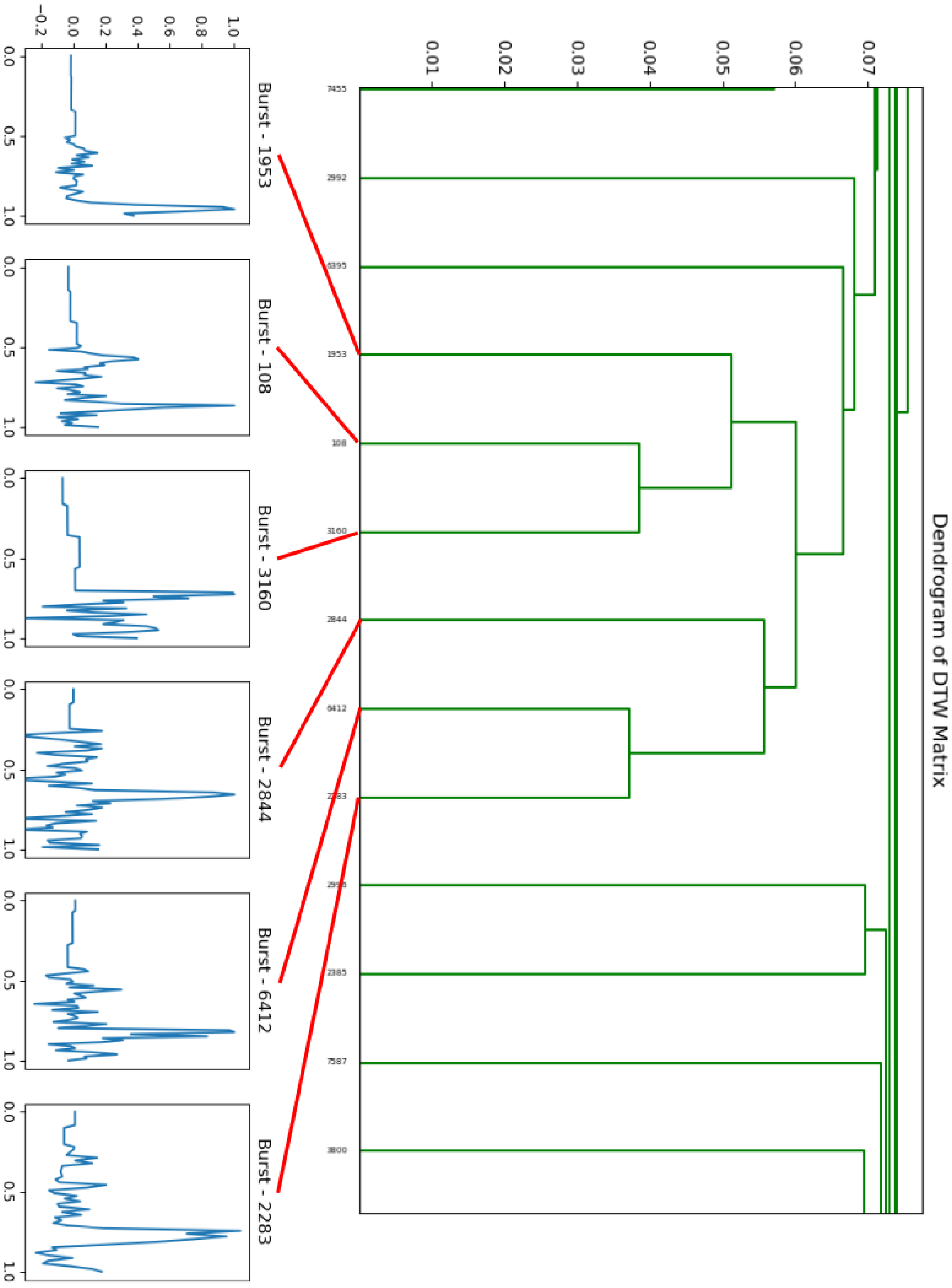

Figure 17 – Emission Cluster Mapping - 1

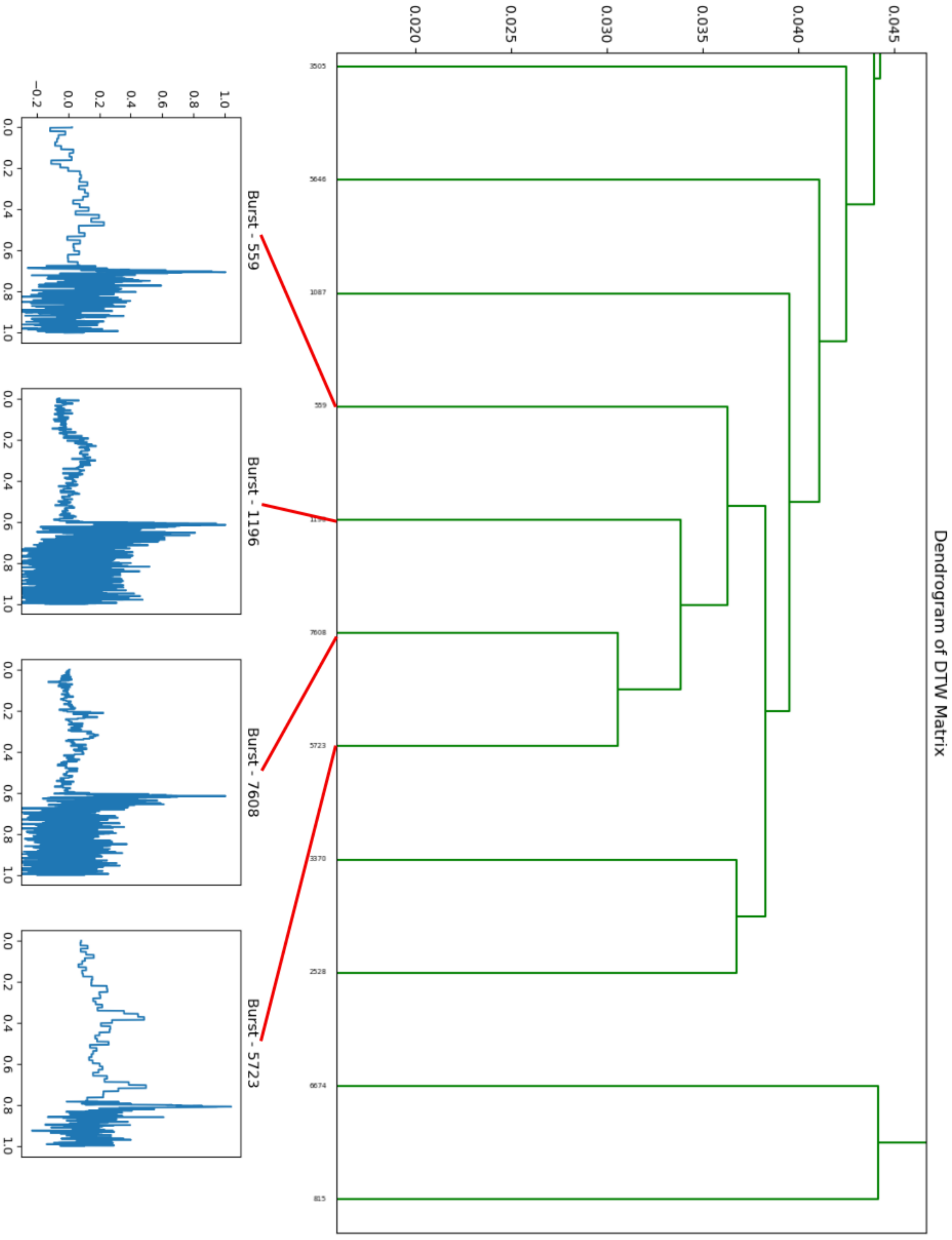

Figure 18 – Emission Cluster Mapping - 2

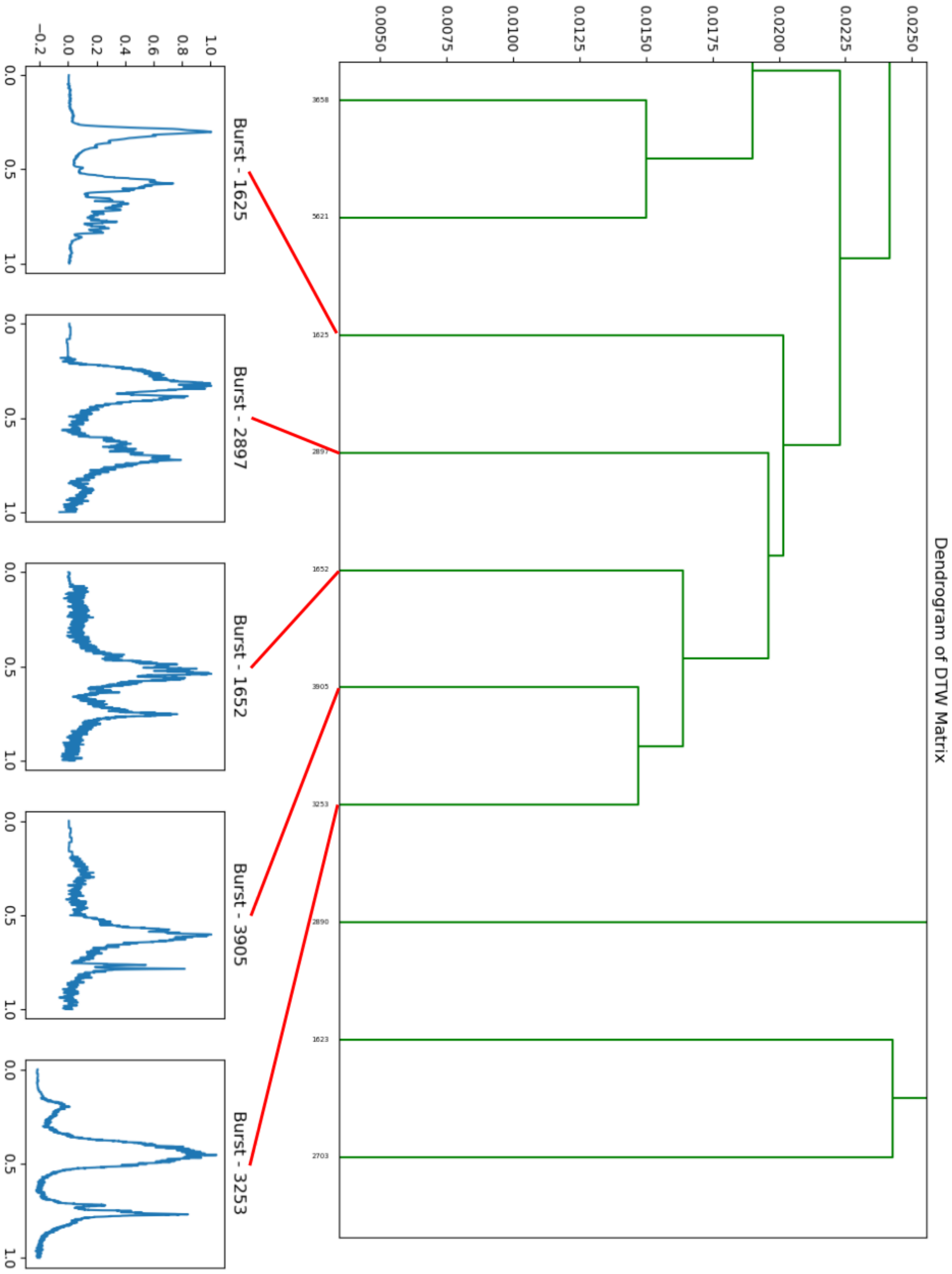

Figure 19 – Emission Cluster Mapping - 3

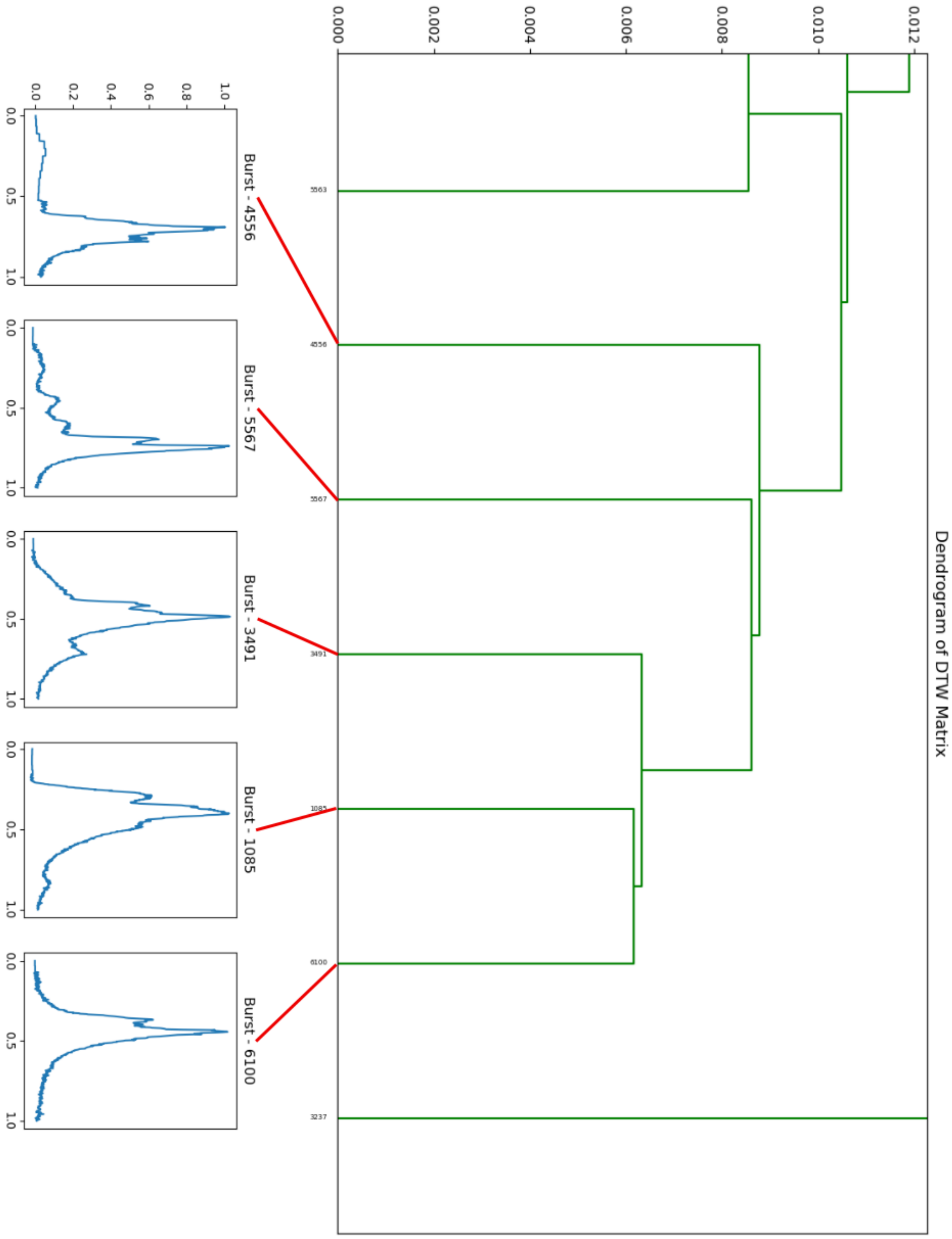

Figure 20 – Emission Cluster Mapping - 4

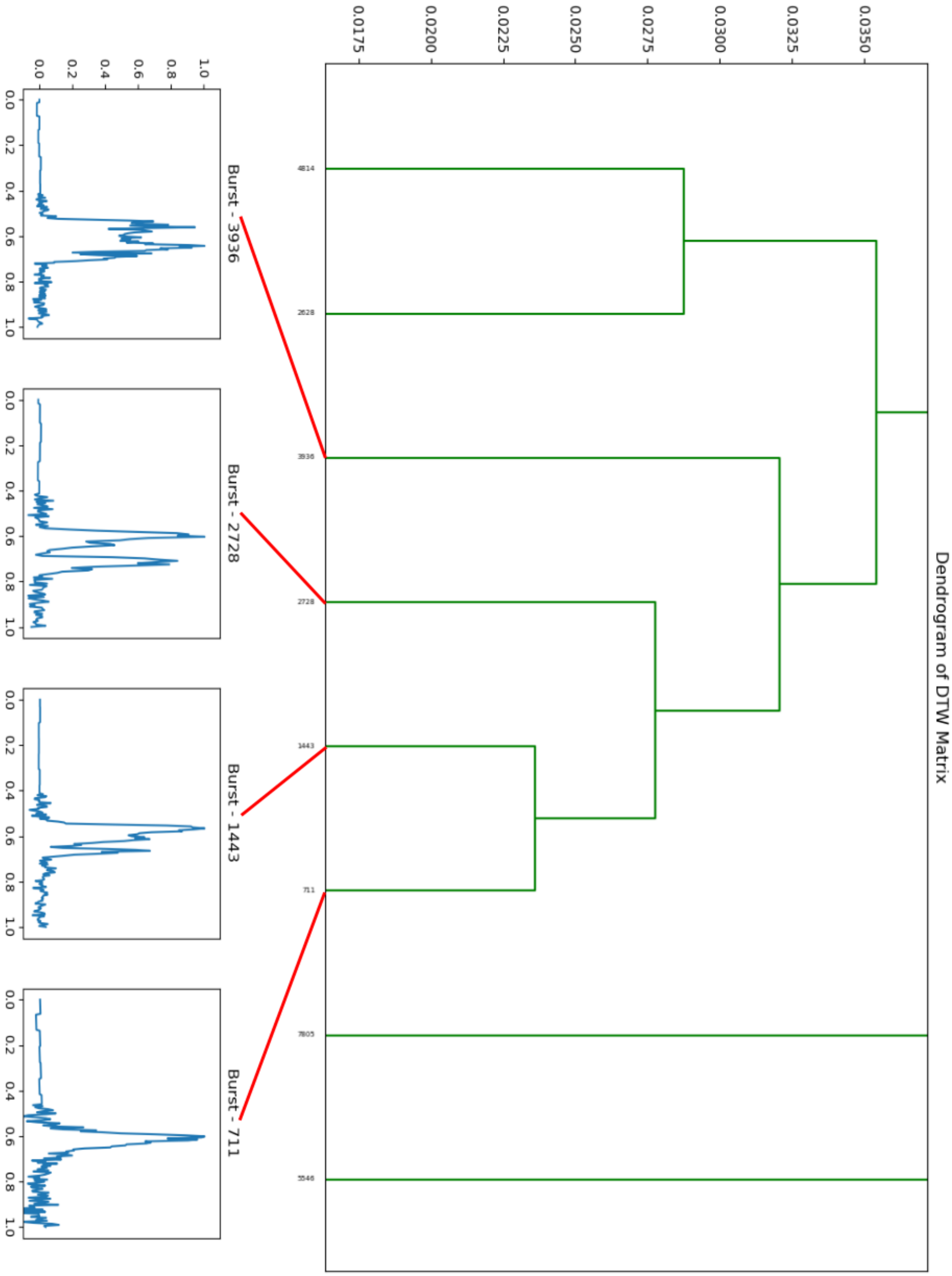

Figure 21 – Emission Cluster Mapping - 5

5.5 Matrix Bias Due to Variable Emission Windows

Through a visual inspection, the cluster developed by DTW outperforms the other methods. The main factor in that performance is its focus on feature correlation over a one-to-one comparison of every element in each vector. This allows the process to be more flexible to any sort of phase shifts that potentially occur between two vectors during preprocessing due to large T90 errors, and a poorly implemented buffer. This causes us to question how important the emission window is and how biased our results may be towards the window for each similarity measure. We do not have an absolute measure of robustness towards these biases; however, we can estimate one by creating two matrices for each measure – one based strictly off of T90 times, and the other with an arbitrary buffer. If a similarity metric is influenced by the addition of non-structural background, then we expect the two matrices to be uncorrelated. On the other hand, if the similarity metric is not influenced by a changing time window, then the matrices should correlate to one another well. Since we believe DTW to not be affected as much by the temporal placement of features, we expect that its two matrices should be more similar than the others.

To test for this, we use the *R* values from a Pearson correlation. Of each one tested, ZNCC evaluated at 0.85, Manhattan at 0.56, and DTW at 0.46. With the lowest *R* value, the changes to the DTW matrices due to adding an arbitrary buffer are the least, serving as a proxy that tells us that DTW is likely a better choice for a similarity metric in our case where we have known errors in the T90 times, which translates to error in the emissions window. While DTW has the lowest R value, besides serving as a comparison to other similarity metrics, we have no threshold on the R value to determine if that information is useful for DTW as a standalone metric.

5.6 Clusters and S/N

Since this is a physical system that was detected by an instrument susceptible to noise by its own internal process and its environment, it is important to understand if there are any biases in the clusters due to non-physical mechanisms such as S/N. A way of accomplishing this is to correlate any non-physical factors back to the clusters. The way to accomplish this is to treat each leaf of the dendrogram as its own cluster and use the value assigned to it by the clustering method that tells us the height at which that leaf broke off of its most relevant parent cluster. These height values correlated with S/N can be seen in Figure 22. A log-log plot of the dendrogram heights and S/N as plotted below has a Pearson R value of -0.79, indicating a negative correlation between the two, meaning, that the more the noise contained in the emission relative to the signal, the more difficult of a time that the clustering process had at finding a leaf's next nearest cluster.

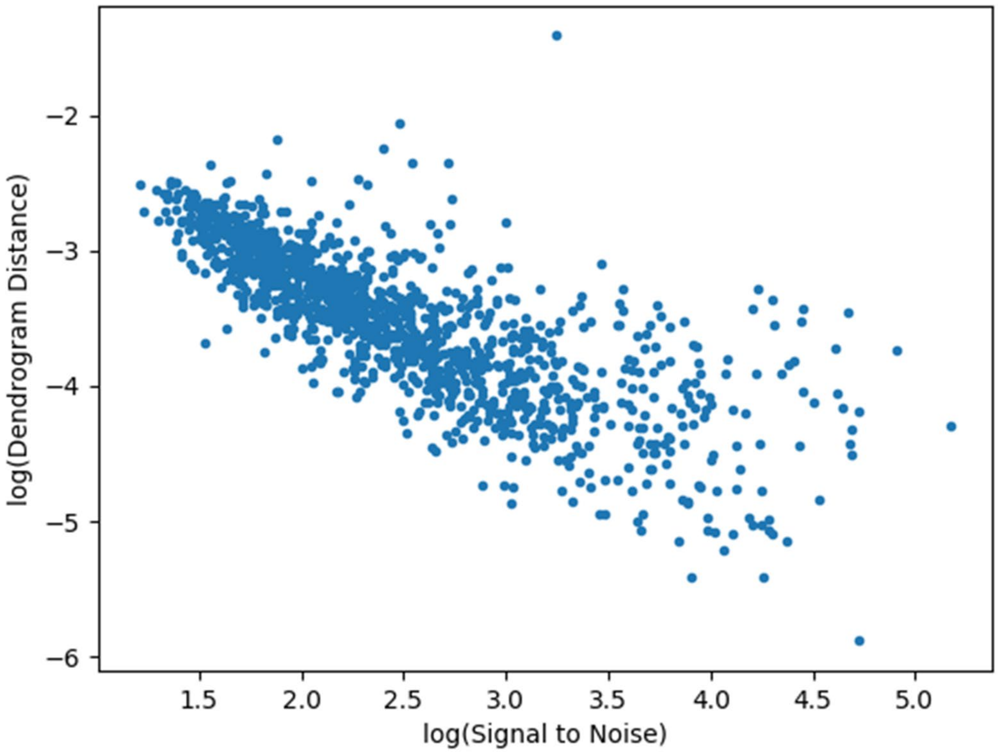

Figure 22 – Dendrogram Distance vs S/N

6. CONCLUSION

We are still only beginning to discover what this data mining technique is teaching us about GRB light curves. Currently, this technique allows us to compare GRB light curves in a new and interesting way. We demonstrated the difficulty in selecting a similarity measurement technique, discussing the S/N and temporal biases that are intrinsic to each emission, and explaining how these biases do not make the analysis a straightforward exercise. We developed preprocessing habits with the data that allowed us to successfully compare all of these emissions despite the biases.

After building similarity matrices with each of the four techniques, we qualitatively evaluated them for correctness through a visual inspection, finding that DTW performed far better than the others and highlighting the strengths and weaknesses of the others. While the DTW matrix performed the best, it still has a bias towards noise, as we showed when we described that the height of a cluster's most recent split is likely a function of S/N. We speculate that any similarity matrix would give this result until further research can be done to correct this bias. It was also shown that because of DTW's algorithmic architecture, the matrix is more robust towards changes in the emission windows.

In astronomy, there is a habit of observing events or structures from similar phenomena, but not understanding the physics behind them well enough to be able to explain what is happening. Clustering has historically served as one of the tools used to help explain how the different events and structures are related (*i.e.* galaxy and star classification). Now, imagine a machine that can create GRB emissions. The machine has all sorts of levers that can be manipulated to create a light curve. Orienting the levers – which represent different physical variables – in one way or another will produce different shapes in the light curves. The ultimate goal is to understand what each lever contributes to the production of the light curve. In order to begin chipping away at this problem, we first need a map of the GRB light curves that tells us how each one is related to the other. By understanding the relationships between the clusters of light curves and the continuum created from the light curve properties, we can begin to pinpoint the underlying physics that is created by the different lever configurations. What we have accomplished in this pursuit is the first step of creating a map of GRB light curves and their observable similarities.

7. FUTURE WORK

We will be working towards a better method of preprocessing the data. One of the major hurdles in the analysis was deciding where to start and stop the emission. The T90 times served well enough to show good results, but they still exhibited errors that trickled down the pipeline. The problem in determining the emissions true window is tied up in S/N. In many emissions, the noise dominates the signal to the point where making an accurate measurement of the emission window is incredibly difficult.

If we can develop a better way to extract the emission window, then perhaps it can also be applied to GRBs with multiple emission episodes. This would potentially add hundreds more samples to the dataset.

We can also attempt to confirm the number of clusters from agglomerative clustering with a different method such as K-means – now that we have an idea of how many clusters exist. A similar ratio of emissions among clusters created by a different clustering method would serve as a validation that the clusters are real. Another future validation could come in the form of a k-folds analysis on the sample with the number of clusters.

REFERENCES

Berndt, D. & Clifford,J. 1994, Using dynamic time warping to find patterns in time series, in: KDD Workshop, Seattle, vol. 10, 359–370

Faloutsos et al. 1994, Fast subsequence matching in time-series databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 419–429

Fishman, G.J. 1992, Gamma-ray Bursts - Observations, Analyses, and Theories, 265

Golenetskii, S. V., Mazets, E. P., Aptekar, R. L., & Ilinskii, V. N. 1983, Nature, 306, 451

Hakkila, J., 2020, unpublished manuscript

Hakkila, J., et al. 2003, ApJ, 582, 320

Hakkila, J., et al. 2008, ApJ, 677, L81

Hakkila, J., et al. 2015, ApJ 815.2, 134

Hakkila, J., et al. 2018, ApJ 863.1, 77

Hakkila, J., & Cumbee, R. S. 2009, in AIP Proc. 1133 (ed. Meegan, Gehrels, & Kouveliotou), 379

Hakkila, J., & Preece, R. 2011, ApJ, 740, 104

Hakkila, J., & Preece, R. 2014, ApJ, 783, 2

Igleisas, F., Energies 2013, 6, 579-597; doi:10.3390/en6020579

Day, W.H.E., Edelsbrunner, H. 1984, Efficient algorithms for agglomerative hierarchical clustering methods. Journal of Classification 1, 7–24, https://doi.org/10.1007/BF01890115

Jain, A. K., Murty, M. N., & Flynn, P. J. 1999. Data clustering: A review. ACM Computing Surveys, 31(3), 264–323

Kate, R. 2016, Using dynamic time warping distances as features for improved time series classification, Data Min. Knowl. Discov, 2, 283–312

Keogh, E. 2002, Exact Indexing of Dynamic Time Warping. In Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, 20–23 August, 406–417

Klebesadel, R. W., Strong, I. B., & Olson, R. A. 1973, ApJ, 182, L85

Kouveliotou, C., Meegan, C. A., Fishman, G. J., Bhat, N. P., Briggs, M. S., Koshut, T. M., Paciesas, W. S., & Pendleton, G. N. 1993, ApJ, 413, L101

Lewis J. 1995, Fast normalized cross-correlation. Vision Interface, 10, 120–123

Liang, E., & Kargatis, V. 1996, Nature, 381, 49

Liao, T.W. 2005, Clustering of time series data—A survey. Pattern Recognition, 38, 1857–1874

Łuczak M. 2016, Hierarchical clustering of time series data with parametric derivative dynamic time warping. Expert Syst Appl, 62, 116–130

Mallozzi, R. 2001, BATSE Instrument Description, July, 2020, https://gammaray.nsstc.nasa.gov/batse/instrument/batse.html

Meegan, C. A., Fishman, G. J., Wilson, R. B., Horack, J. M., Brock, M. N., Paciesas, W. S., Pendleton, G. N., & Kouveliotou, C. 1992, Nature, 355, 143

Mukherjee S., et al. 1998, ApJ, 508, 314

Norris, J. P., Nemiroff, R. J., Bonnell, J. T., Scargle, J. D., Kouveliotou, C., Paciesas, W. S., Meegan, C. A., & Fishman, G. J. 1996, ApJ, 459, 393

Norris, J. P. 2002, ApJ, 579, 386

Norris, J. P., Bonnell, J. T., Kazanas, D., Scargle, J. D., Hakkila, J., & Giblin, T. W. 2005, ApJ, 627, 324

Paciesas et al. 1999, The Astrophysical Journal Supplement Series, 122, 465–495

Paciesas et al. 1996, 4B Gamma-Ray Burst Catalog (revised), July, 2020, https://gammaray.nsstc.nasa.gov/batse/grb/catalog/4b/

Paczynski, B. 1991, Acta Astron., 41, 257

Ramirez-Ruiz, E., & Fenimore, E. E. 2000, ApJ, 539, 712

Rodgers, J.L. 1988, Nicewander, W.A. Thirteen ways to look at the correlation coefficient. Am. Stat., 42, 59–66

Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. 2011, The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30

van Paradijs, J., et al. 1997, Nature, 386, 686

Virtanen, P. et al. 2020, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, 261-272

Wang, Xiao & Yu, Fusheng & Pedrycz, Witold & Wang, Jiayin 2019, Hierarchical clustering of unequal-length time series with area-based shape distance, Soft Computing. 23, 6331- 6343, 10.1007/s00500-018-3287-6

Yoo J-C, Han T. 2009, Fast Normalized Cross-Correlation. Circuits, Systems, and Signal Processing, 28, 6, 819–843.

APPENDICES

A GitHub repository exists that contains all files and code needed to recreate this project. It is available at this link:

https://github.com/twcannon/MastersThesis

Contents:

\data\

Contains all pickled matrix files, burst lists, and linkage files. Also contains the *background_table.csv*, *burst_info.csv*, and *duration_table.csv*

\Paper\

Contains all raw image files shown in the paper and the paper itself.

\GRBCluster\

Contains all of the scripts needed to run the project.