**TITLE**

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

**MASTER OF SCHENCE**

**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:**

ABSTRACT

**TITLE**

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

**MASTER OF SCHENCE**

**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’s) 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 in 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

Test

CONTENTS

[ABSTRACT i](#_Toc45742024)

[ACKNOWLEDGEMENTS iii](#_Toc45742025)

[CONTENTS iv](#_Toc45742026)

[LIST OF FIGURES v](#_Toc45742027)

[1. INTRODUCTION 1](#_Toc45742028)

[1.1 Historical Context 1](#_Toc45742029)

[1.2 GRB Emissions 2](#_Toc45742030)

[1.3 Objectives 3](#_Toc45742031)

[1.4 BATSE 64ms Data 4](#_Toc45742032)

[1.5 Summary 6](#_Toc45742033)

[2. SIMILARITY MEASURES FOR TIME SERIES DATA 7](#_Toc45742034)

[2.1 Euclidean Distance 7](#_Toc45742035)

[2.2 Zero-Normalized Cross-Correlation 8](#_Toc45742036)

[2.3 Dynamic Time Warping 9](#_Toc45742037)

[2.4 Normalized Manhattan 10](#_Toc45742038)

[3. AGGLEMERATIVE HIERARCHICAL CLUSTERING 12](#_Toc45742039)

[4. METHODS 14](#_Toc45742040)

[4.1 Preprocessing 14](#_Toc45742041)

[4.2 Constructing the Matrices 15](#_Toc45742042)

[4.3 Clustering 15](#_Toc45742043)

[5. RESULTS 17](#_Toc45742044)

[6. CONCLUSION 18](#_Toc45742045)

[7. FUTURE WORK 19](#_Toc45742046)

LIST OF FIGURES

Test

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 were 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 can be seen in Figure **GET A 2X2 PIC OF BURSTS AS EXAMPLES**. We define an emission episode as an increase in the gamma ray flux above the background noise with no discernable relationship to any other adjacent increase in flux above the background. 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.(1996)). 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)). Unfortunately, in a highly structured GRB emission episode, when a burst is comprised of what seems to be a large number of overlapping pulses, 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, GBR pulses exhibit residual fluctuations in phase with the pulse structure 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, but they 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). 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 Figure **GET A HISTOGRAM OF LONG/SHORT** (Kouveliotou et al.(1993); Mukherjee et al.(1998); Hakkila et al.(2003)). 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 bursts 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.

## 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 subjectively looks to the observer as the same phenomena. One of the most well-known examples that comes to mind 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 see the population as a whole.

## 1.4 BATSE 64ms Data

BATSE (Fishman(1992)) Figure **GET A PIC OF BATSE** 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 was collected in four different energy channels, ranging from highly energetic X-rays to gamma rays. A GRB can vary in its emission throughout each channel, and in some cases 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.

The satellite orbited earth in an elliptical orbit, which plummeted the experiment in an out of earth’s radiation belts. The radiation belts contributed to the background noise in the detectors, the level of which can be easily 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 (eg. Vela X-1), and gamma ray producing black holes (eg. Cyg X-1). Besides background noise sources and 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.

The data used for this study is archived from the life of the experiment. The summed four channel data has a time resolution of both 256ms and 64ms per burst. The BATSE experiment collected data at 256ms 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, some GRB samples in our working dataset will have both of the time resolutions along their light curves.

The data is freely available to download as ASCII files. The files contain a few lines of meta-data with information such as the count of bins of the burst and the total number bins 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 bin. 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 bin.

We will also be using 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 flux if the burst is observed. We use these time frames 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. Despite the data being normalized, there are still biases we will potentially be carrying over into the analysis from the raw data. These biases will be mitigated through the steps of the clustering process. Three of the largest steps are data preprocessing, building an adequate similarity matrix, and choosing the proper clustering routine. Several different methods between preprocessing and building matrices were attempted, 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 will describe how the steps of preprocessing the data and the application of the defined methods for producing the similarity matrices and clustering. Section 5 will discuss the biases, strengths, and weaknesses for select permutations of the pipelines from preprocessing to cluster results. It will also discuss the results themselves and what it means for GRB physics.

1. SIMILARITY MEASURES FOR TIME SERIES DATA

Binned Time-series data is a sequence of real numbers representing the total counts of an event per a given increment of time. We consider a similarity measure as a resemblance value that is calculated between any two vectors 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 (Igleisas (2013)).

In order to construct a similarity matrix, we need to generate a value of resemblance between two vectors of every GRB to every other GRB, 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 and of length , 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 and vectors to create our Euclidean similarity measure . Thus, it is represented by

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 invariant 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 is not normalized between every pair of vectors, 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.

## 2.2 Zero-Normalized Cross-Correlation

As stated above, Euclidean distance is blind to the correlation between features. 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 to 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 the pairs 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.

Assuming vectors of equal length, we select the max value from ZNCC as the distance measure, giving

ZNCC attempts to fix the problem of blindness to feature correlation that the Euclidean method has as well as the problem or normalization of similarity measures between multiple pairs of vectors. 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, where and . DTW builds an -by- cost matrix that contains the distances between two points and . Then a warping path is formed by the set of matrix components, where . In addition, the warping path should satisfy three local constraints:

1. Endpoint constraint: and
2. Monotonicity constraint: if and

then and ;

1. Continuity constraint: if and

then and .

See Figure **INSERT DTW GRAPHIC** for an illustration of the distance geometry and Figure **INSERT DTW GRAPHIC** for an illustration of the optimal warping path. 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

## 2.4 Normalized Manhattan

One last similarity test that is used as a simple calculation for compare 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 :

1. AGGLEMERATIVE 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, having each entry 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 you see the single, complete, and average methods. 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 are going to 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 you were to average all the GRB emissions episodes in a given branch, you 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 used the python SciPy package to conduct the clustering.

IDL routine called CLUSTER\\_TREE.\footnote{ For the IDL clustering routine used see \url{http://www.exelisvis.com/docs/cluster\\_tree.html}}\footnote{For the analogous MATLAB routine see \url{http://www.mathworks.com/help/stats/linkage.html}}\footnote{For the analogous Python Scipy routine see \url{http://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html}}

1. 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 is a raw dataset and if rife with irregularities that cause some data to be unusable for our purposes and need to be filtered out. There were three large filters that every burst had to pass through.

The first filter was to ensure 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 data set 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 form 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 **CITE THIS** 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. This was done because our assumption is 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 is 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 **CITE THIS** 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 and more would cause 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 need 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 bins up to a larger amount would not work, because we would be creating data that does not necessarily exist. We therefore would resample emissions of a larger number of bins down to match the emission with a smaller number of bins. In order to prevent the resampling of hundreds of bursts down to a number of bins where the structure of the data gets washed out by resampling to a much smaller number of bins, we decide 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 familiar to the GRB community.

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 has a sloped or changing background that needs 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 changes 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 is no function that can robustly fit it. We therefore result to using a simple linear model on the background noise located on the flanks of each emission.

Using the T90 times, we can find the middle 90% of the emission. A buffer is added to either side of the middle 90% to mask the entire emission. A margin is then continued out from either side where the buffer left off. This margin is what creates two segments of data that is comprised solely of background noise surrounding the emission that will be used to calculate the background slope.

The background is quickly calculated with *scipy.stats.linregress* **CITE THIS** , which returns 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 begin creating the similarity matrices by using the data from emissions that have been zeroed and leveled. From here, the pipeline has two paths it can continue down. We need to compare only the emission episode, so the adjacent background must be trimmed away. However, trimming by the T90 times will truncate 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 cannot test this concept now, we can likely estimate how well using T90 times would work by comparing clustering results form T90 times with clustering results of a time window that encompasses the whole emission, as discussed later.

The second path that the pipeline can possibly take is to add a buffer to the T90 times to ensure that the entire emission is visible. Since we are assuming that all GRBs originate from similar physical processes regardless of their duration, we can construct the buffer as a function of the T90 durations. To create the buffer, we simply extend 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 give us two different windows to build similarity matrices from.

Now that the raw data has been zeroed, flattened, and had its windows defined, we now must ensure that the vectors are the same length and of comparable scale. We determine 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* **CITE THIS** function. Then the scale of the vectors is normalized from 0 to 1 by simply dividing by the max of each vector. This give 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 use the four methods described above. These four methods plus the two options of using the T90 buffer or not give 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.

## 4.3 Clustering

We continue with the SciPy python package and use *scipy.cluster.hierarchy.linkage* **CITE THIS** function on the data from the pickled python files. As described above, we use the average linkage method. We also use the *optimal\_ordering* flag, which for a few more seconds of processing time, will organize the data so that any future dendrogram produced from it will exhibit a more intuitive tree structure. The *optimal\_ordering* flag does not any of the clustering process itself.

We produce a dendrogram with the *scipy.cluster.hierarchy.dendrogram* function. The two necessary inputs are 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 **PUT DENDROGRAM HERE.** It 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 **PUT SMALLER DENDROGRAM HERE** .

1. RESULTS

With the clustering completed, we move the data into a dendrogram for easy visualization. With our large dataset, the dendrogram becomes too large to visualize on a single page. Therefore, as seen in Fig. 1, we sample a part of the dendrogram for preliminary results discussion. The dendrogram visualizes different GRB light curves as leaves in branches. Each branch represents a cluster of GRBs that have similar characteristics as seen in Fig. 2. The vertical length of a branch is a measure of how similar a GRB or cluster of GRBs is to its connecting GRB or cluster of GRBs. Sometimes, the tree will display several adjacent GRBs in a tight cluster that show evidence for an evolving continuum of GRB properties as seen in Fig. 3. Also, There are some GRBs 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 GRBs would have been selected last as a comparable relative to any other GRB or cluster of GRBs. Curiously, these GRBs also have an incredibly high signal to noise.

1. CONCLUSION

Hakkila et al. (2018) asks, “As signal-to-noise decreases, does structure disappear before or after the smoothly-varying remainder of the pulse disappears?”

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. By using a two-sampled $\chi^2$ test results as `similarity distances' in a Agglomerative Hierarchical Clustering Algorithm, we are able to achieve our comparisons. This technique seems to be good at finding GRBs that evolve similarly and possibly have similar progenitors. However, with a large amount of new data to investigate, we are still determining the meaning of the results of the analysis. We are also working on ways to improve and speed up the computationally expensive process.

1. FUTURE WORK

While the method is sound the data preparation could be improved upon. A better definition of the burst start and end times would work nicely. Also, a more rigorous analysis needs to be done with the results. We have developed a new tool with which to explore GRB physics. However, in the larger scheme of things, this method does not have to be used simply for GRB light curves. It is easily adaptable to any transient or two dimensional data.

**REFERENCES**

BATSE data <http://www.batse.msfc.nasa.gov/batse/grb/catalog/current/>

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

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

[Fishman(1992)]{1992grbo.book..265F} Fishman, G.~J.\ 1992, Gamma-ray Bursts - Observations, Analyses and Theories, 265

[Golenetskii et al.(1983)]{gol83} Golenetskii, S. V., Mazets, E. P., Aptekar, R. L., \& Ilinskii, V. N. 1983, Nature, 306, 451

[Hakkila et al.(2003)]{hak03}Hakkila, J., Giblin, T. W., Roiger, R. J., Haglin, D. J., Paciesas, W. S., \& Meegan, C. A. 2003, ApJ, 582, 320

[Hakkila et al.(2008)]{hak08}Hakkila, J., et al. 2008, ApJ, 677, L81

[Hakkila \& Cumbee(2009)]{hak09}Hakkila, J., \& Cumbee, R. S. 2009, in AIP Proc. 1133 (ed. Meegan, Gehrels, \& Kouveliotou), 379

[Hakkila \& Preece(2011)]{hak11} Hakkila, J., \& Preece, R. 2011, ApJ, 740, 104

[Hakkila \& Preece(2011)]{hak11} 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. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification* **1,**7–24 (1984). 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., Using dynamic time warping distances as features for improved time series classification, Data Min. Knowl. Discov. 2 (2016) 283–312.

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

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

[Kouveliotou et al.(1993)]{kou93}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. Fast normalized cross-correlation. Vision Interface. 1995;10:120–123.

[Liang \& Kargatis(1996)]{lai96}Liang, E., \& Kargatis, V. 1996, Nature, 381, 49

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

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

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 et al.(1998)]{muk98}Mukherjee S., et al. 1998, ApJ, 508, 314

[Norris et al.(1996)]{nor96}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(2002)]{nor02}Norris, J. P. 2002, ApJ, 579, 386

[Norris et al.(2005)]{nor05}Norris, J. P., Bonnell, J. T., Kazanas, D., Scargle, J. D., Hakkila, J., \& Giblin, T. W. 2005, ApJ, 627, 324

Paciesas et al. THE ASTROPHYSICAL JOURNAL SUPPLEMENT SERIES, 122:465–495, 1999 June

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

[Ramirez-Ruiz \& Fenimore(2000)]{ram00}Ramirez-Ruiz, E., \& Fenimore, E. E. 2000, ApJ, 539, 712

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

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

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. Fast Normalized Cross-Correlation. Circuits, Systems, and Signal Processing. 2009;28(6):819–843.

**FIGURES**

Test

**APPENDICES**

Test