Unsupervised Machine Learning for Analysis of Young Star Photometric Light Curves

Karen Zhou

Mentor: Professor Lynne Hillenbrand

Abstract

Time series data such as stock market prices in finance or light curves in astronomy present a notoriously challenging problem in machine learning. Spectral clustering is a type of unsupervised machine learning that transforms data points into a graph structure where each point is a node and edges represent the similarity between nodes. We apply spectral clustering to the normalized flux of light curves in the Upper Scorpius and Taurus clusters in the K2 and TESS datasets to obtain a nuanced understanding of stellar variability and characteristics beyond those available using standard lightcurve metrics. We utilize quantile graphs for dimensionality reduction, capturing the general motion of the light curves while remaining robust to attributes such as amplitudes, period, and phase as well as fluctuations in the data. By applying an unsupervised method to a traditionally supervised task, we aim to uncover novel insights in the light curves.

Background

In astronomy, there is an abundance of data on countless objects such as stars and galaxies as well as observational techniques including imaging, photometry spectroscopy, polarimetry, and time domain surveys. Particularly in the present day, it becomes increasingly difficult to analyze this wealth of data and gain insight from it, en masse. However, machine learning may facilitate this process of discovery by streamlining the process of finding connections between data points with many features.

While there is a great deal of work with supervised learning in astronomy from Rodríguez-Feliciano et al.'s optimization algorithm for characterizing light features to Mistry et al.'s work with random forest classifiers for cataclysmic variables, the application of unsupervised learning algorithms is relatively unexplored in comparison[1][2]. In Professor Hillenbrand's work and that of those working with her, they have pioneered groundbreaking research in the field of observational astrophysics, particularly focusing on the formation and evolution of young stellar objects. Through meticulous analysis of optical infrared and

submillimeter data as well as the development and application of innovative observational techniques, Professor Hillenbrand and her team have unraveled key insights into the early stages of star formation, shedding light on the physical processes governing these celestial phenomena, as well as the properties of young accretion disks and older debris disks around stars.

Each light curve within the K2 and TESS datasets from their corresponding NASA missions is rich in features, presenting a valuable opportunity to enhance the performance and robustness of unsupervised machine learning algorithms. As depicted in Figure 1, light curves of young stars often exhibit a degree of complexity and non-periodicity, which is particularly noticeable in the TESS dataset. This intricacy underscores the potential benefits of employing unsupervised machine learning techniques for analysis, as traditional visual inspection methods may be insufficient to fully capture the nuances present in these datasets. By leveraging the multitude of features available in each data point, unsupervised machine learning holds promise in uncovering underlying patterns and structures in the data that may not be readily discernible through manual examination alone.

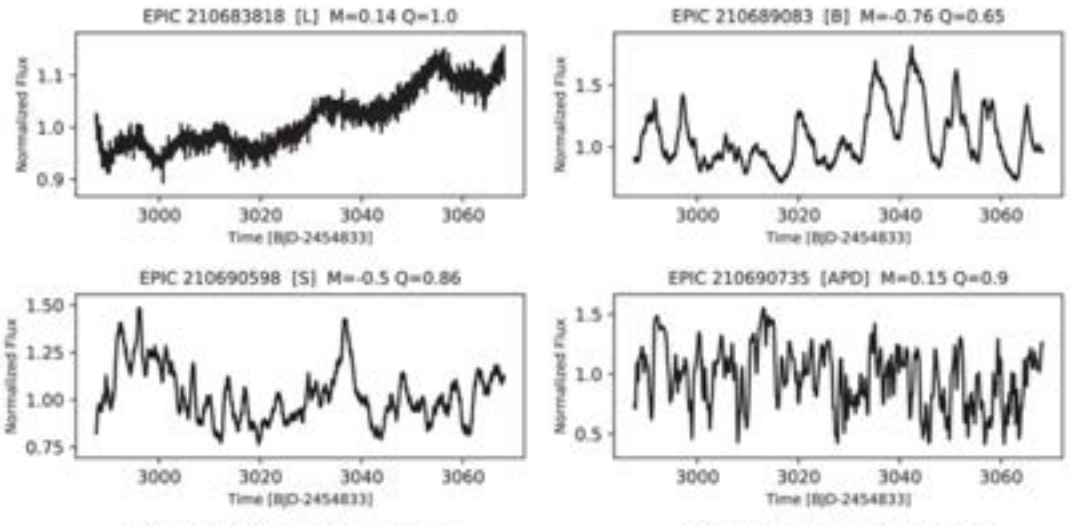


Fig. 1. Light curves for disk-bearing and non-disk-bearing sources towards Taurus. Young star light curves are difficult to gain insight from because of their inherent disorder[3].

In Ann Marie Cody et al.'s previous work, the light curves in the Kepler/K2 dataset were categorized into eight variability classes based on their periodicity and symmetry, as illustrated in Table 1[3]. Thus, in this project, we aimed to derive new insights from cluster young stellar object

groups with a focus on investigating the related and defining characteristics that make them a group. The K2 dataset played a crucial role in this analysis, offering exhaustive data with which we could explore the intricacies of these stellar formations[4]. While the TESS dataset offers a wealth of intriguing information, it is important to note the periodic discontinuities in the data attributed to the 2-minute sampling cadence for the photometric data and the 30-minute cadence for reading the full image frame[5]. Consequently, the initial application of unsupervised algorithms will focus on the K2 dataset due to its more consistent sampling and longer duration dataset, though covering fewer objects of interest. We had observed previously that the K2 dataset is suitable for supervised machine learning applications, as demonstrated by a previous SURF student[6].

Name	Symbol	Name	Symbol
Aperiodic Dipper	APD	Quasi-periodic Dipper	QPD
Burster	B	Quasi-periodic	QPS
Long-timescale	L	Periodic	P
Multiperiodic	MP	Stochastic	S

Table 1. The eight variability classes and their corresponding symbols[3].

One of the primary obstacles to achieving greater understanding was the intrinsic disorder found in young star light curve data. This disorder often stems from factors such as variability in accretion rates, extinction events, and the intricate interplay of stellar and circumstellar dynamics[7]. For the most part, light curves display periodic behavior, a characteristic well-suited for analysis using machine learning techniques. However, aperiodic behavior remains relatively unexplored, mainly due to the conventional reliance on visual inspection for analysis. Utilizing unsupervised machine learning offers valuable insights into this less understood aspect of light curve analysis. Previous studies have identified behavior linked to the rotation of the star or circumstellar disk, which can be classified as periodic or quasiperiodic using Fourier techniques like periodograms, as seen in figure 2[3]. Additionally, the presence of aperiodic behavior due to accretion and extinction variability has been observed; however, quantifying this phenomenon

has proven exceedingly challenging. Despite these difficulties, recent advancements in data analysis methods showed promise in unraveling these complex phenomena.

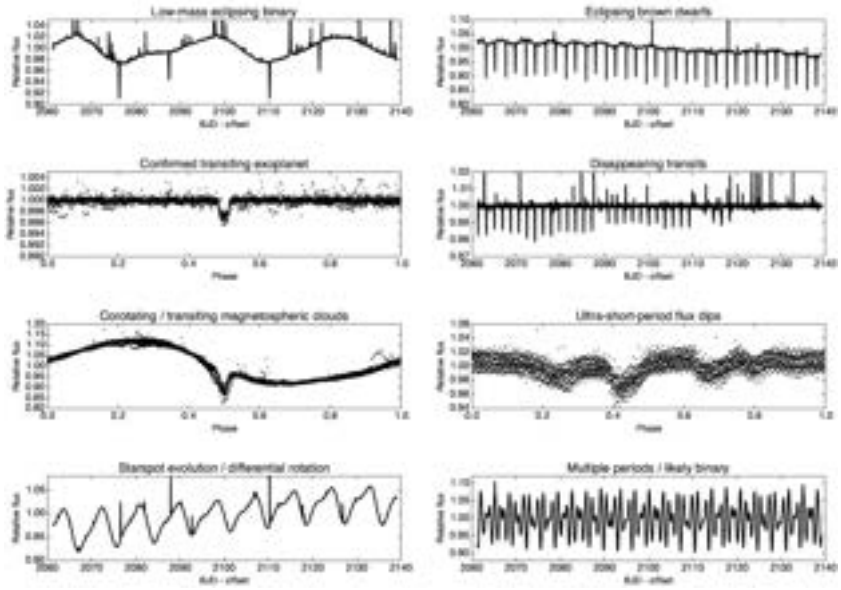


Figure 2. Sample light curves from the Kepler/K2 dataset displaying periodic behavior[3].

Ultimately, we successfully separated bursters and dippers into distinct clusters, clearly demonstrating the effectiveness of our approach for these particular classes of young stars. However, the classification of other stellar behaviors, such as stochastic and periodic variables, remains an area for further investigation. While the initial results are promising and provide significant insights into the clustering of light curves, we recognize the need for improved data visualization techniques to more effectively uncover and interpret the subtle underlying patterns within the dataset. By enhancing the visualization method, we hope to get a more comprehensive understanding of the complex interactions and variations present in the light curves. Additionally, we hope to extend the application of our developed and modified techniques from the K2 dataset to the TESS light curves. By doing so, we aim to validate our findings across different datasets and uncover new insights that may further contribute to the characterization and understanding of young stellar objects.

Method

We have focused on analyzing the Upper Scorpius cluster of stars in the K2 dataset. This cluster contains 288 light curves, each with detailed data points including the date, cadences,

flux, uncertainty, x-position, y-position, and quality. Additionally, each light curve is linked to an EPIC identification number and includes attributes like primary and secondary variability (if applicable), amplitude, timescale, additional period, Q, and M values. Our main focus has been on utilizing the flux variance over time, but we have also made use of EPIC IDs, date, variability class, period, Q, and M values to determine the success of the method. While we have begun examining the Taurus cluster in K2, the primary testing ground remains the Upper Scorpius cluster to refine our techniques.

Throughout this project, we utilized Google Colab as the computational environment and imported several key libraries, including NumPy, pandas, scikit-learn, Matplotlib, Seaborn, and UMAP, for data processing and visualization. Additionally, we employed the glob module to retrieve all relevant light curve files from the specified directory.

Data Processing

To standardize the data, we began by reading all the light curve files, which contain individual data points in rows, and storing the dates and median-normalized flux in separate arrays with corresponding indices. Next, we ensured uniformity across the arrays by trimming any extra points beyond the length of the light curve with the fewest data points, followed by plotting the light curves as a sanity check as seen in figure 3. Finally, we removed the first sixty points from each light curve to eliminate any noise associated with the initial recording phase.

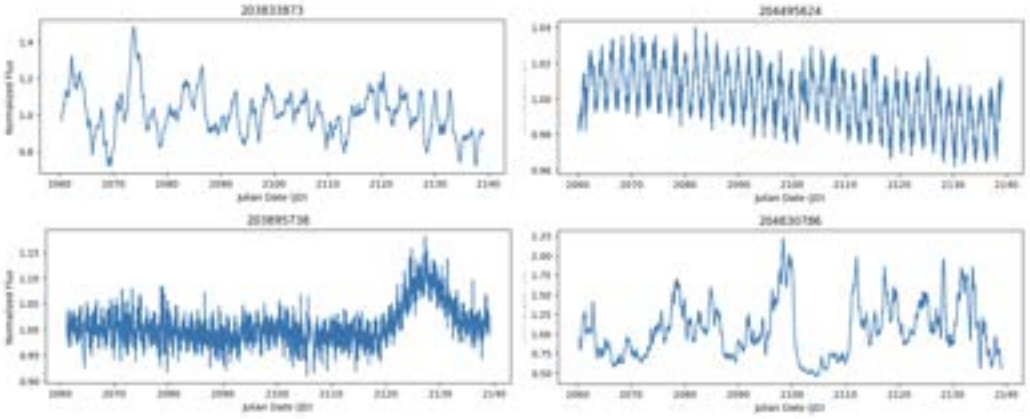


Fig. 3. Light curves from the K2 dataset graphed with normalized curves.

Then, we utilized min-max normalization to normalize the data (eq. 1). By transforming the normalized flux values to be between 0 and 1, this method preserves the relationships

between the original data points and ensures that all features contribute equally to the analysis, enhancing the performance of clustering algorithms. On the other hand, when we tried z-score normalization, there was far too much variance given the large range that the data spans (eq. 2). Thus, scaling all the values to match a range rather than using the mean ensures that the entire shape of each light curve can be analyzed, and the numerous outliers do not skew the clustering results.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{1}$$

$$X' = \frac{X - \mu}{\sigma}, \tag{2}$$

where X_{min} and X_{max} are the global minimum and maximum and μ is the mean and σ is the standard deviation.

Quantile Graphs

One of the challenges in applying machine learning to light curves is the inherent variability in their appearance, which arises from phase shifts, varying periods, and differing amplitudes. This variability can lead to discrepancies in how light curves are interpreted by algorithms compared to human astronomers. For example, two light curves that an astronomer might classify as similarly periodic could be rated as dissimilar by an algorithm like k-nearest neighbors. This issue and its implications for our analysis will be explored in greater detail later in the discussion. This is where quantile graphs come into play, offering a method to address these inconsistencies by preserving the underlying patterns of the light curves despite their variations.

Using the resulting normalized flux values, we reduced the dimensionality from 3,329 points to 49 with the help of quantile graphs, which map each transition between quantiles from point to point. Quantile graphs, a "simple method for transforming a time series into graphs and networks", retains relevant properties of the original time series in the topological features of the resulting graph by converting a teme series $X = \{x(t) | t \in \mathbb{N}, x(t) \in \mathbb{R}\}$ into a complex network $g = \{N, A\} \in G$ for N nodes or vertices and A edges or arcs[8]. Two nodes are connected with a weighted edge whenever two values $x(t)$ and $x(t + k)$ belong to quantiles q_i and q_j for some

$t = 1, 2, \ldots, T$ and $k = 1, \ldots, k_{max} < T$ where T is the duration of the time series, and thus repeated transitions through the same edge will increase the weight corresponding to the edge, as seen in figure 4[8]. This graph of transitions can then be stored in an adjacency matrix, which is a $Q \times Q$ matrix with rows corresponding to recorded q_i values and columns corresponding to recorded q_j values, and we can perform our calculations on these resulting matrices.

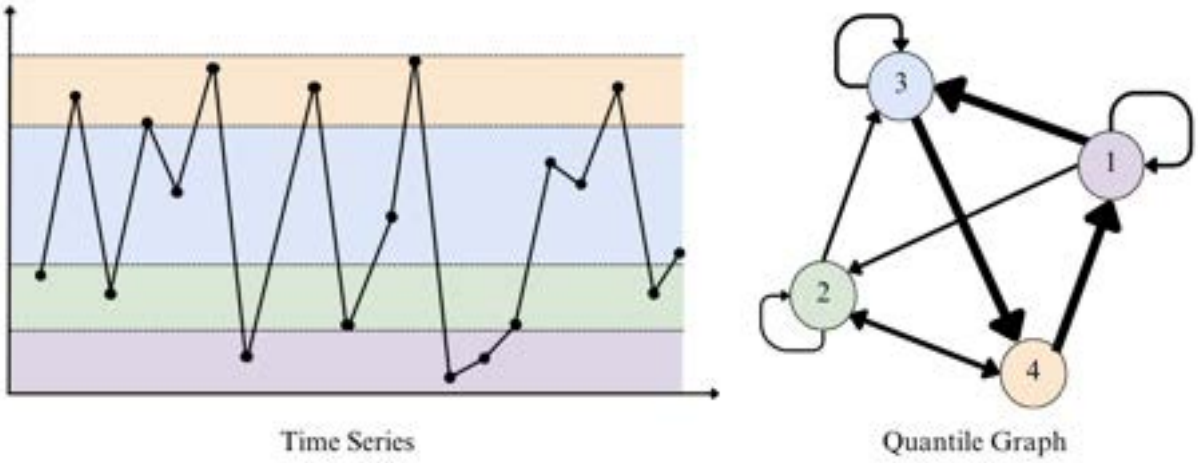


Fig. 4**.** Quantile graph for a time series where Q = 4[9].

Beginning with $Q = 7$, we calculated the quantiles based on the values of the entire min-max normalized dataset by aggregating the values and then finding the quantiles using the *quantile* method in NumPy. For each time series, we determined the transitions between quantiles and stored these transitions in the adjacency matrix, a two-dimensional array where the first index corresponds to the quantile location of the initial value in the time series and the second index corresponds to the subsequent quantile location. After constructing the resulting adjacency matrix, we flattened it into a one-dimensional array. All of these flattened arrays were then stacked to form a single two-dimensional array with row indices corresponding to each respective EPIC identification number that had been stored earlier. This method effectively preserves the overall motion of the light curve while remaining robust against potential phase shifts while ensuring that we could still identify each individual data point.

We tested various values of k (1, 2, 3, 4, 5, 7, 10, 15, and 20) to determine the optimal parameter for our analysis. We ultimately found that a k value of 3 yielded the best results, as it provided a balanced overview without excessively focusing on individual samples. Higher k

values, particularly those above 5, led to a diminished clarity of individual features, causing the patterns observed in the quantile graphs to dissipate (see Appendix A).

Additionally, we evaluated different methods for calculating the quantiles. While our final approach of calculating quantiles across the entire dataset proved highly effective, we also experimented with calculating quantiles individually for each light curve. This approach was unsuccessful, as it caused the resulting quantile graphs to appear overly similar, rendering the clustering algorithm unable to effectively distinguish between the different light curves. Consequently, we opted to calculate quantiles on the entire dataset to maintain the distinctiveness of each light curve in the clustering process, as seen in figure 5 (see Appendix B).

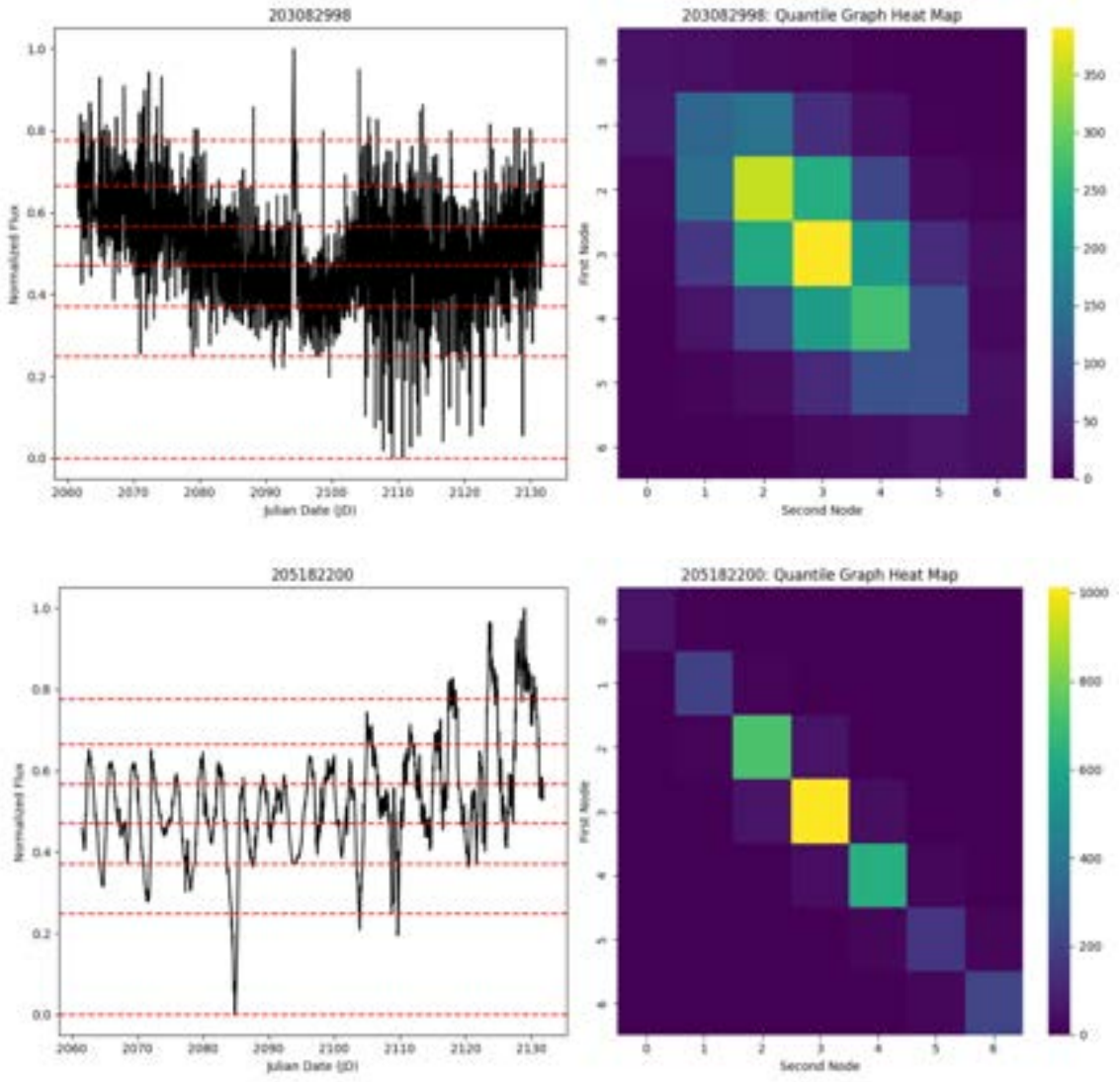


Fig. 5. Sample light curves on the left and their corresponding quantile graphs on the right. Calculated quantiles are graphed on the light curve in red.

Unsupervised Clustering

The final algorithm we chose to use and have been optimizing is spectral clustering, which is effective for identifying clusters in data by transforming it into a graph structure and using eigenvalues to find groupings. We determined this algorithm to be the best option compared to k-means, hierarchical clustering, and DBSCAN, as seen in figure 6. K-means, a method that partitions data into a predefined number of clusters by minimizing within-cluster variance, performed poorly because it assumes that there exists clusters of equal variance. Hierarchical clustering, which starts with all data points in one cluster and recursively splits them in a non-agglomerative approach, was slightly more promising than k-means, but it is likely that its sensitivity to noise and outliers distorted cluster formation. Lastly, DBSCAN, which forms a cluster if there are a minimum number δ of points within a specified distance ε from a starting point, labeled most of the data as noise and only formed a single cluster containing every light curve. We attempted to adjust the parameters, as these can significantly impact the outcomes; however, after exploring a ε range of [0.5, 1.5] and a δ range of [3, 5], the results remained unsatisfactory.

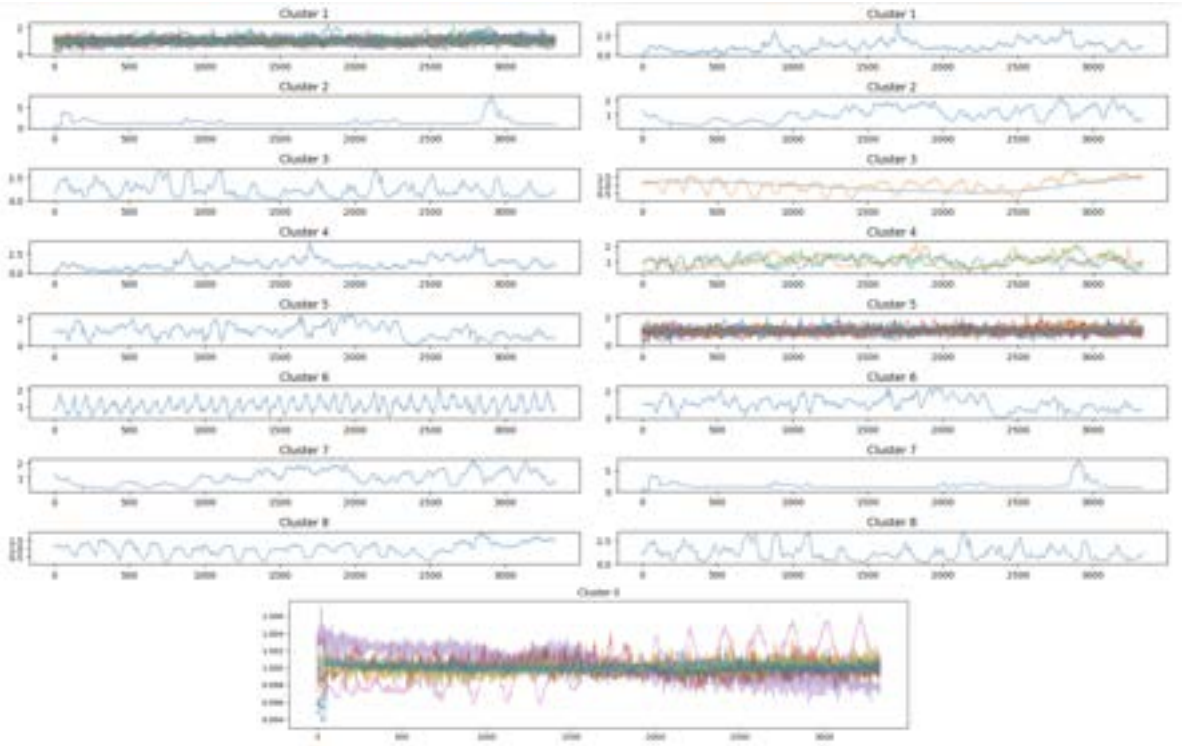


Fig. 6. K-means clustering results on the left, hierarchical clustering results on the right, and DBSCAN clustering results on the bottom.

Spectral clustering is a graph-based algorithm that maps each data point to a lower-dimensional space where clusters can be identified more easily. While computationally intensive for larger datasets, it handles the few hundred light curves analyzed at present without issue. The process begins by constructing a similarity matrix and then generating a normalized Laplacian matrix L_{sym} (eq. 3). This matrix undergoes eigenvalue decomposition, where the eigenvalues and eigenvectors are computed. The smallest nonzero eigenvalues are used to project the data into a lower-dimensional space, forming a new set of points that reflect the structure captured by the Laplacian matrix:

$$L_{sym} = I - D^{-\frac{1}{2}} A D^{\frac{1}{2}} , \tag{3}$$

where I is the identity matrix, D is the degree matrix, and A is the adjacency matrix. A is not to be confused with the adjacency matrix for the quantile graphs, which is where the adjacency matrix is built by selecting edges based on quantile thresholds to focus on specific connections, and the adjacency matrix for the Laplacian matrix represents all or a standard set of connections without necessarily applying quantile based filtering. The process of calculating the Laplacian thereby facilitates the identification of clusters using a standard clustering algorithm such as k-means.

Spectral clustering is particularly advantageous for clustering young star light curves due to its ability to capture complex, non-linear structures that are often present in astronomical data. Unlike traditional clustering algorithms that may struggle with irregularly shaped clusters, spectral clustering leverages the eigenvalues of a similarity matrix to reveal the underlying geometry of the data while preserving the integrity of the temporal patterns. This method is effective for light curves because it can discern subtle variations and patterns that are not easily captured by other algorithms. The similarity matrix, which forms the core of spectral clustering, encapsulates the relationships between different light curves based on their proximity in feature space, making it ideal for handling the nuanced differences in the periodicity and amplitude of young star light curves. Thus, spectral clustering enables the identification of clusters that align with the intrinsic characteristics of the light curves.

During testing, we systematically evaluated various parameters to optimize the performance of the spectral clustering algorithm (see Appendix C). In the end, we employed eight clusters, corresponding to the eight known variability classes. To construct the similarity

matrix, I utilized the k-nearest neighbors algorithm, setting the number of neighbors (*n_neighbors*) to 8 and the number of initializations (*n_init*) to 8. The remaining parameters were left at their default settings, as they have been pre-optimized for efficiency, effectiveness, and stability—particularly the use of the Arnoldi algorithm for eigenvalue computation.

The k-nearest neighbors (KNN) method I employed for constructing the similarity matrix uses the Minkowski distance with $p = 2$, equivalent to the Euclidean distance. However, this metric can cause light curves that should be grouped together to appear dissimilar due to phase shifts and period variations inherent in light curve sampling, as seen in figure 7. To address this issue, we introduced quantile graphs for dimensionality reduction, which effectively capture the overall movement of the light curves while mitigating the impact of these variations and other anomalous behaviors.

Fig. 7. Light curves that would be ranked dissimilar by KNN despite clear similarity.

Cluster Analysis and Visualization

We used UMAP to verify that the adjacency matrices brought greater structure to the distribution of the light curves, facilitating the application of machine learning techniques. UMAP works by constructing a high-dimensional graph of the data and then optimizing its low-dimensional representation to preserve both local and global data structure. Compared to PCA, which linearly transforms the data by maximizing variance along orthogonal axes, UMAP is better suited for this context because it captures complex, non-linear relationships within the data, as seen in figure 8. Nevertheless, we also plotted PCA results to compare the effectiveness of these dimensionality reduction methods, as seen in figure 9. For each adjacency matrix or light curve, we color-coded them according to their corresponding variability class labels. However, when UMAP was applied to adjacency matrices, the original color sorting seen in the light curves was less pronounced, suggesting that the adjacency matrix representation may obscure some of the intrinsic clustering patterns captured in the raw light curve data. This

discrepancy likely arises because the adjacency matrices emphasize strong local relationships, which can alter the global structure that UMAP detects.

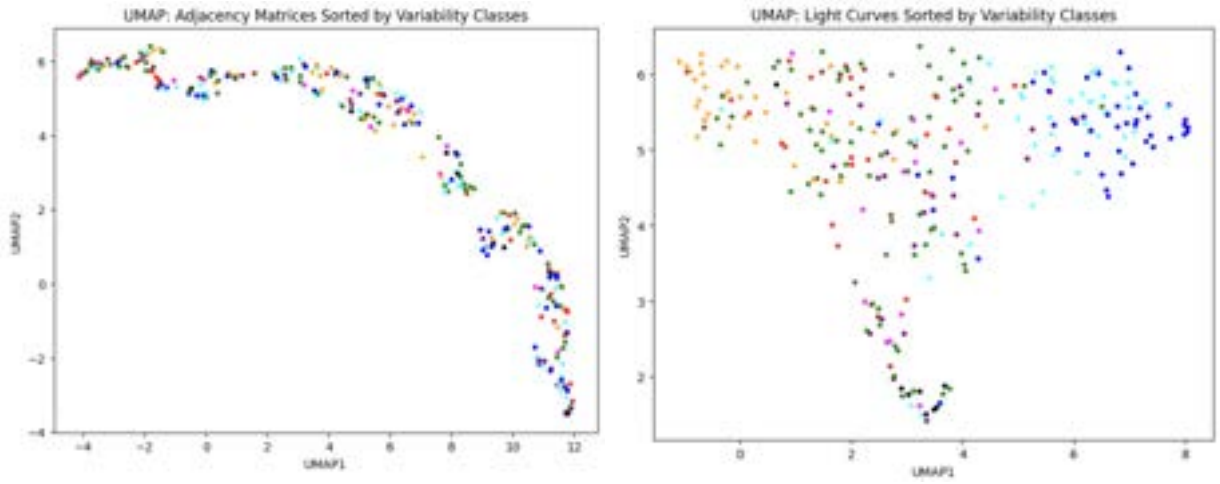


Fig. 8. UMAP results color coded by variability class. Results using adjacency matrices on the left, original light curves on the right.

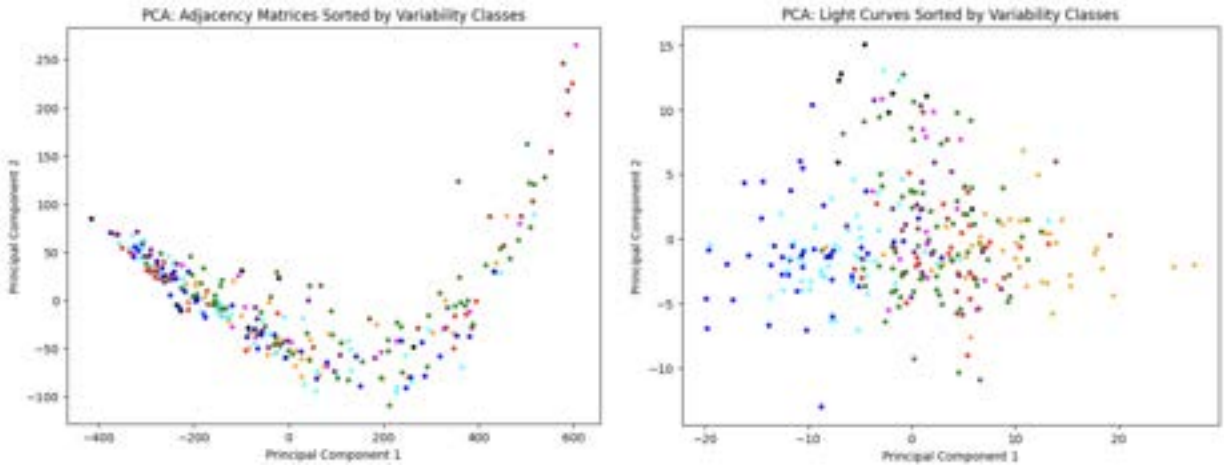


Fig. 9. PCA results color coded by variability classes. Results using adjacency matrices on the left, original light curves on the right.

In the eight clusters sorted by variability classes, we observed a clear tendency for the clustering algorithm to assign aperiodic dippers and bursters to distinct clusters, as seen in figure 10. This separation highlights the algorithm's ability to differentiate between these two types of variability, which are often associated with unique physical processes in young stars. The distinct

clustering of these types suggests that the algorithm effectively captures the underlying features that set aperiodic dippers and bursters apart from other light curves.

Interestingly, we also found that aperiodic dippers frequently clustered together with quasiperiodic dippers, while multiperiodic and quasiperiodic symmetric light curves tended to group within the same clusters. This behavior suggests that the algorithm identifies shared characteristics between these variability classes, which might reflect commonalities in their underlying physical mechanisms. However, long-timescale and stochastic light curves did not show a strong presence in any particular cluster, indicating that these types may be more diverse or less tightly defined by the features used in the clustering process.

Additionally, we noticed that light curves classified as type N, U, and QP, categories outside the original eight variability classes, were mostly assigned to a single cluster as seen in figure 11. This observation raises the possibility of an underlying pattern or relationship among these outliers that could be significant. While these clustering results might initially seem surprising, they point to patterns in the data that could be important to understand further. Although there is a chance these findings stem from nuances in the clustering algorithm, our thorough testing suggests that these results are noteworthy and warrant deeper investigation.

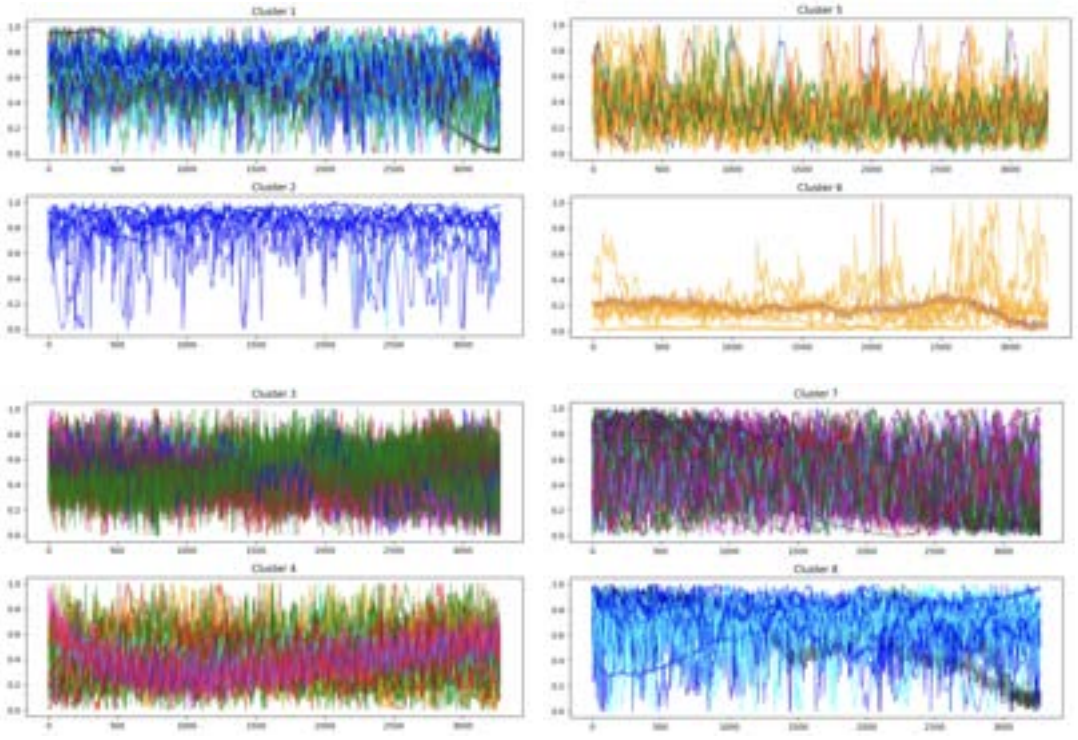


Fig. 10. Spectral clustering for the Upper Scorpius light curves color coded by variability class.

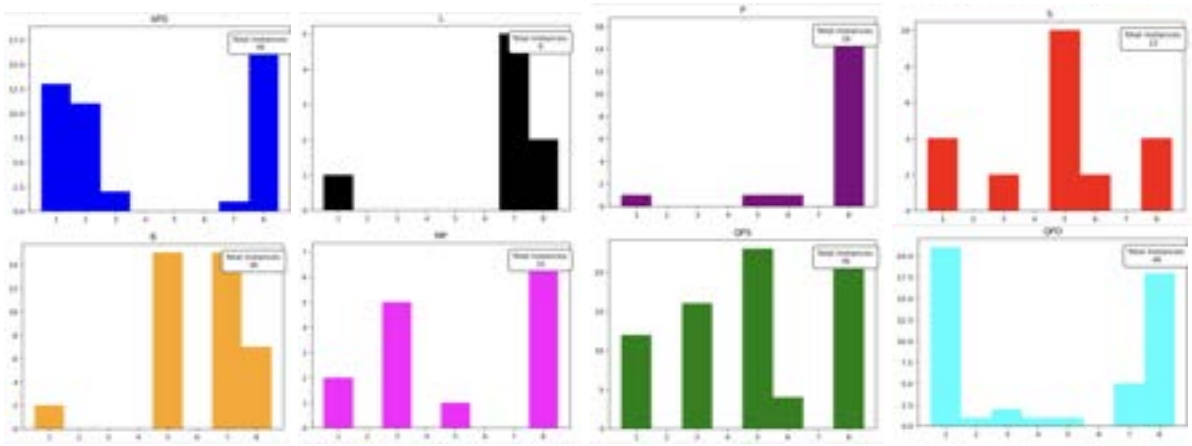
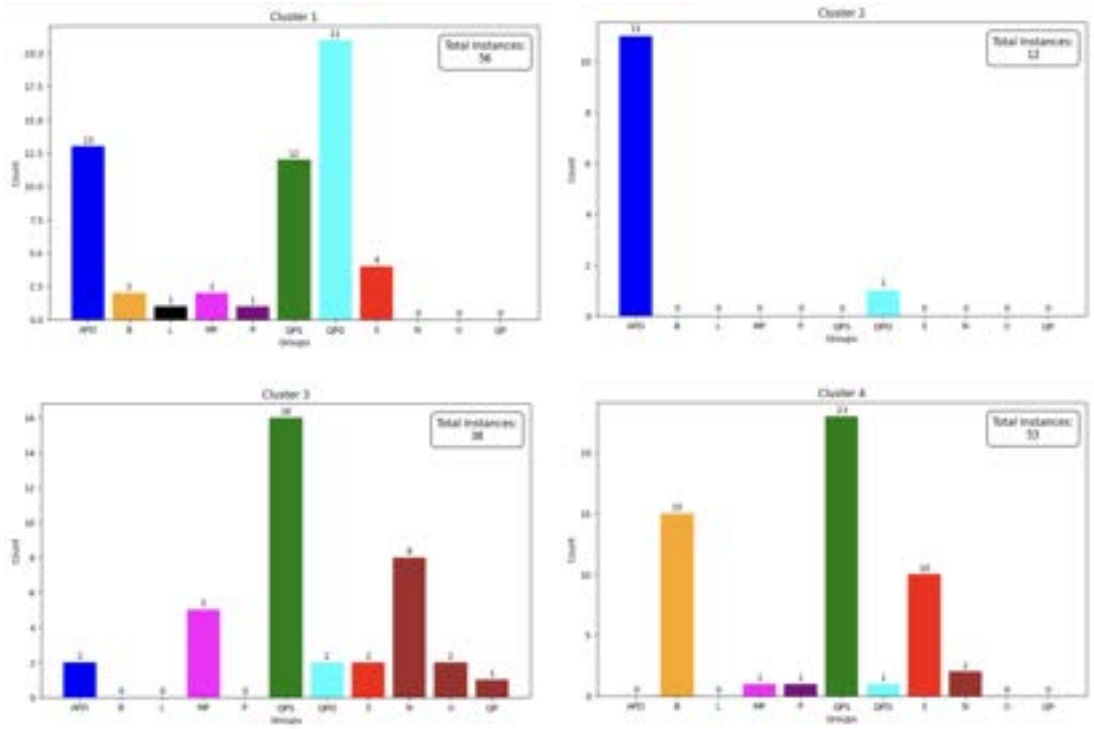


Fig. 11. Distribution of cluster labels for each of the variability classes.

These findings suggest a need for further exploration of the clustering algorithm's behavior and the features it emphasizes. Refining the algorithm or integrating additional features could reveal more subtle patterns in the light curves, enhancing our understanding of variability classes beyond those found in figure 12. Additionally, the clustering of outlier types (N, U, and QP) may point to a new classification or an overlooked aspect of young star behavior, warranting further investigation.



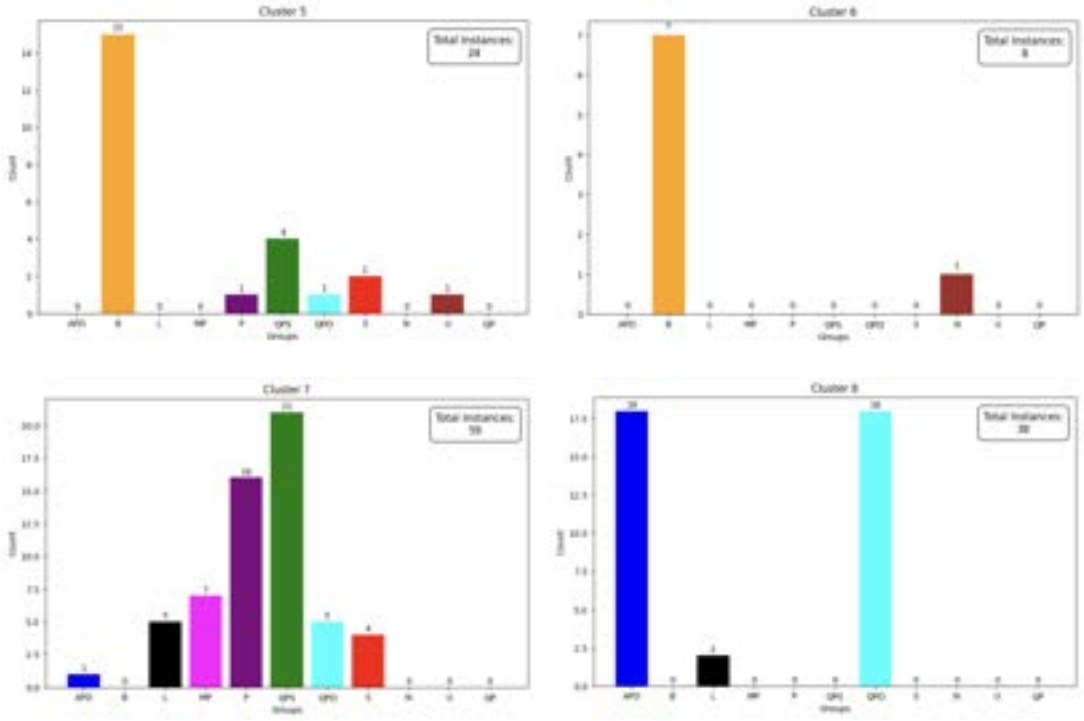


Fig. 12. Distribution of variability class for the clusters.

It is likely that the burster and dipper light curves were frequently grouped together due to their characteristic behaviors, where the majority of flux values are concentrated in either the first or last quantiles, with only occasional spikes or dips. This steady movement within the light curves may have led to their consistent clustering. However, despite these observations, the clusters remain somewhat disorganized, indicating a need for improved data visualization methods to enhance interpretability. One potential approach to address this challenge would be to plot the quantile graphs corresponding to different clusters, which could provide additional insights into the underlying patterns and relationships within the data.

Extension to Taurus Light Curves

While much of our research has focused on the Upper Scorpius cluster within the K2 dataset, we are also interested in extending our work to the Taurus cluster as a precursor to analyzing young stellar objects in the TESS dataset. A notable distinction within the Taurus cluster is that some objects exhibit more than one variability class. Despite this difference, the application of our methodology to this new set of light curves has yielded even more intriguing

results as seen in figure 13, suggesting that our approach may be broadly applicable across different stellar populations.

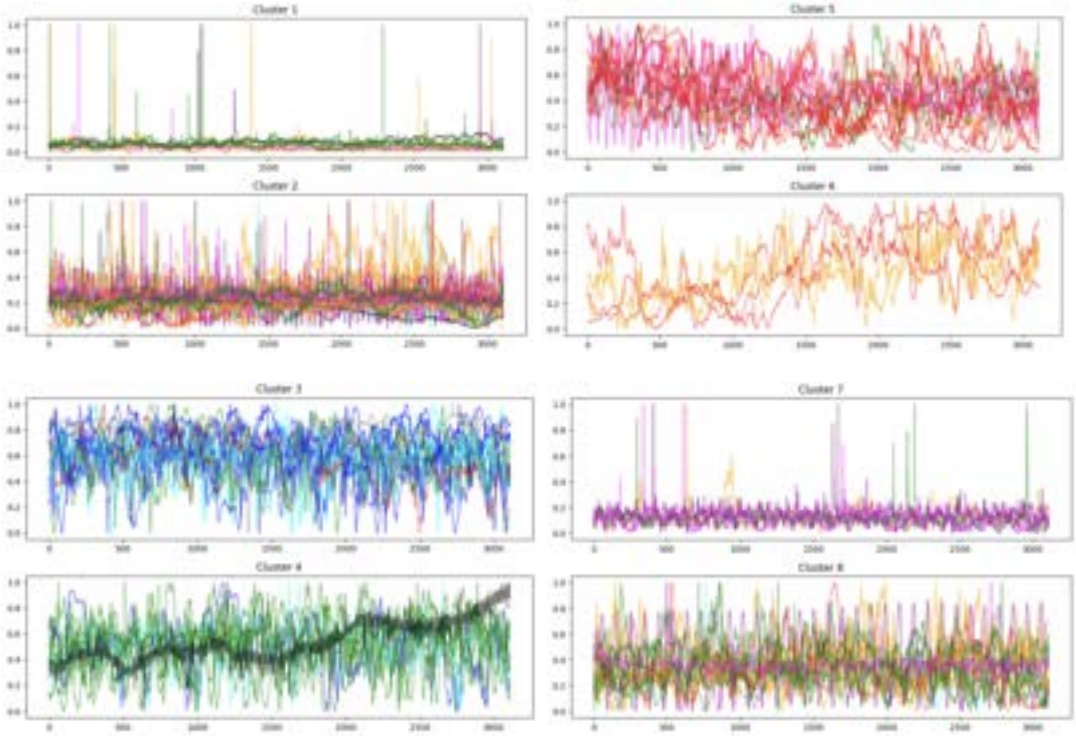


Fig. 13. Clustering on the Taurus light curves using the current method.

The identification of previously unrecognized burster objects is particularly intriguing, as it raises the question of whether the dominant characteristic of these light curves is their overall motion rather than isolated spikes or fluctuations, which may represent anomalies in the data recording process. Even if we account for the possibility of such errors, it is notable that all quasiperiodic symmetric light curves were consistently grouped within the same cluster. Furthermore, when examining the overlaid light curves, it becomes evident that many light curves within the same cluster exhibit similar motion patterns, reinforcing the effectiveness of the clustering methodology.

Conclusion

In conclusion, we analyzed K2 light curve data from the Upper Scorpius and Taurus regions to explore the clustering of young stars. The data was processed using quantile graphs, which allowed us to emphasize the most significant relationships between light curves and

effectively reduce dimensionality. We employed UMAP and PCA to visualize the resulting structures, with UMAP particularly highlighting complex, non-linear relationships within the data. Following this, we applied spectral clustering to the quantile graphs to identify natural groupings within the light curves. The clustering results were then interpreted and validated through detailed data visualization techniques, providing insights into the underlying patterns of variability among the stars in these regions.

The next phase of our research involves applying the methodologies developed for the Taurus stellar object cluster in the K2 dataset to the corresponding group of light curves in the TESS dataset, as seen in figure 14. By leveraging our existing results from the K2 data, we aim to determine whether the outcomes derived from the TESS light curves are consistent and provide additional insights. This comparative analysis will help assess the robustness and generalizability of our approach across different observational datasets.

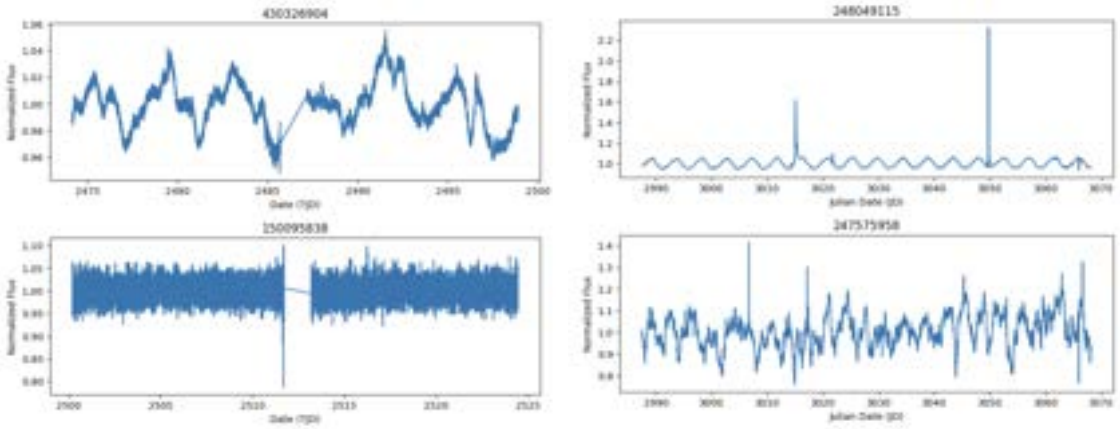


Fig. 14. TESS light curves on the left, K2 light curves on the right. Note the gaps in observation in the TESS light curves.

K2 and TESS are both integral to NASA's ongoing efforts to monitor the sky, utilizing similar technologies and aiming to achieve comparable observational goals, such as detecting exoplanets and exploring stellar variability. Despite their shared objectives, the operational differences between the two missions result in distinct data characteristics. K2's focused monitoring of a smaller sky area allows for longer, continuous observations, which facilitates a more detailed study of stellar phenomena[4]. In contrast, TESS surveys a broader portion of the sky by dividing it into 26 sectors, leading to high-precision data but with observable gaps due to

the satellite's operational constraints[5]. Consequently, while K2 offers more detailed insights into stellar behavior, TESS captures a wider range of objects. Applying the methods developed in this study to TESS data could therefore unlock significant potential, providing astronomers with a deeper, more unified understanding of stellar behavior across different regions of the sky.

Acknowledgments

I would like to express my deepest gratitude to my mentors, Professor Lynne Hillenbrand and postdoctoral scholar Luke Bouma from Caltech for their invaluable guidance and support throughout the research process. Their expertise and dedication have been instrumental in shaping this project, and I am incredibly fortunate to have had the opportunity to learn from them. I would also like to extend my sincere thanks to Professor Dovi Poznansky from Tel Aviv University for his insightful contributions and guidance as a visiting professor. His expertise and support have greatly enriched this research, and I am deeply appreciative of his willingness to share his knowledge.

I would also like to thank the Summer Undergraduate Research Fellowship (SURF) program and the Student-Faculty Programs (SFP) office at Caltech for providing me with the resources and platform to pursue this research. Their commitment to fostering undergraduate research has made this project possible, and I am grateful for the opportunity to be a part of such a rewarding and enjoyable experience.

References

[1] Rodríguez-Feliciano, Bayron et al. “Machine-learning Morphological Classification of TESS Light Curves of T Tauri Stars.” *The Astronomical Journal*, vol. 166, no. 5, 10 October 2023, 10.3847/1538-3881/acf865.

[2] Mistry, Dharmesh et al. “Machine Learning based search for Cataclysmic Variables within Gaia Science Alerts.” 4 October 2022, https://doi.org/10.1093/mnras/stac2760.

[3] Cody, Ann Marie et al. “The Many-faceted Light Curves of Young Disk-bearing Stars in Taurus as Seen by K2.” *The Astronomical Journal*, 13 April 2022, vol. 163, no. 5, 10.3847/1538-3881/ac5b73.

[4] “K2 Information.” *NASA Exoplanet Archive*, 3 November 2021, https://exoplanetarchive.ipac.caltech.edu/docs/K2Mission.html.

[5] “TESS | Documentation.” *Mikulski Archive for Space Telescopes*, https://archive.stsci.edu/missions-and-data/tess#:~:text=TESS%20will%20conduct%20high%2precision,TPFs)%20and%20calibrated%20light%20curves.

[6] Carr, Philip. “Application of Supervised Machine Learning to Classification of Variable Young Stars.” https://sites.astro.caltech.edu/~lah/pcarr_2018SURFreport.pdf.

[7] Findeisen, Krzysztof et al. “Simulated Performance of Timescale Metrics for Aperiodic Light CUrves.” *The Astrophysical Journal*, 10 January 2015, http://dx.doi.org/10.1088/0004-637X/798/2/89.

[8] Pineda, Aruane et al. “Quantile Graphs for EEG-based Diagnosis of Alzheimer’s Disease.” *PLOS ONE*, 5 June 2020, https://doi.org/10.1371/journal.pone.0231169.

[9] Silva, Vanessa et al. “Multilayer Quantile Graph for Multivariate Time Series Analysis and Dimensionality Reduction”. *International Journal of Data Science and Analytics*, 27 May 2024, https://link.springer.com/article/10.1007/s41060-024-00561-6

[10] “SpectralClustering.” *skikit-learn Documentation*, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html.

Appendix A

During the evaluation of quantile graph parameters, we tested various values of k (1, 2, 3, 4, 5, 7, 10, 15, and 20) to determine their impact on capturing the characteristics of the light curves, as seen in figure A1. Our analysis indicated that a k value of 3 was the most effective in representing the overall motion of the light curves while avoiding an excessive focus on local features, ultimately striking a balance between capturing the general trends and mitigating the influence of localized anomalies.

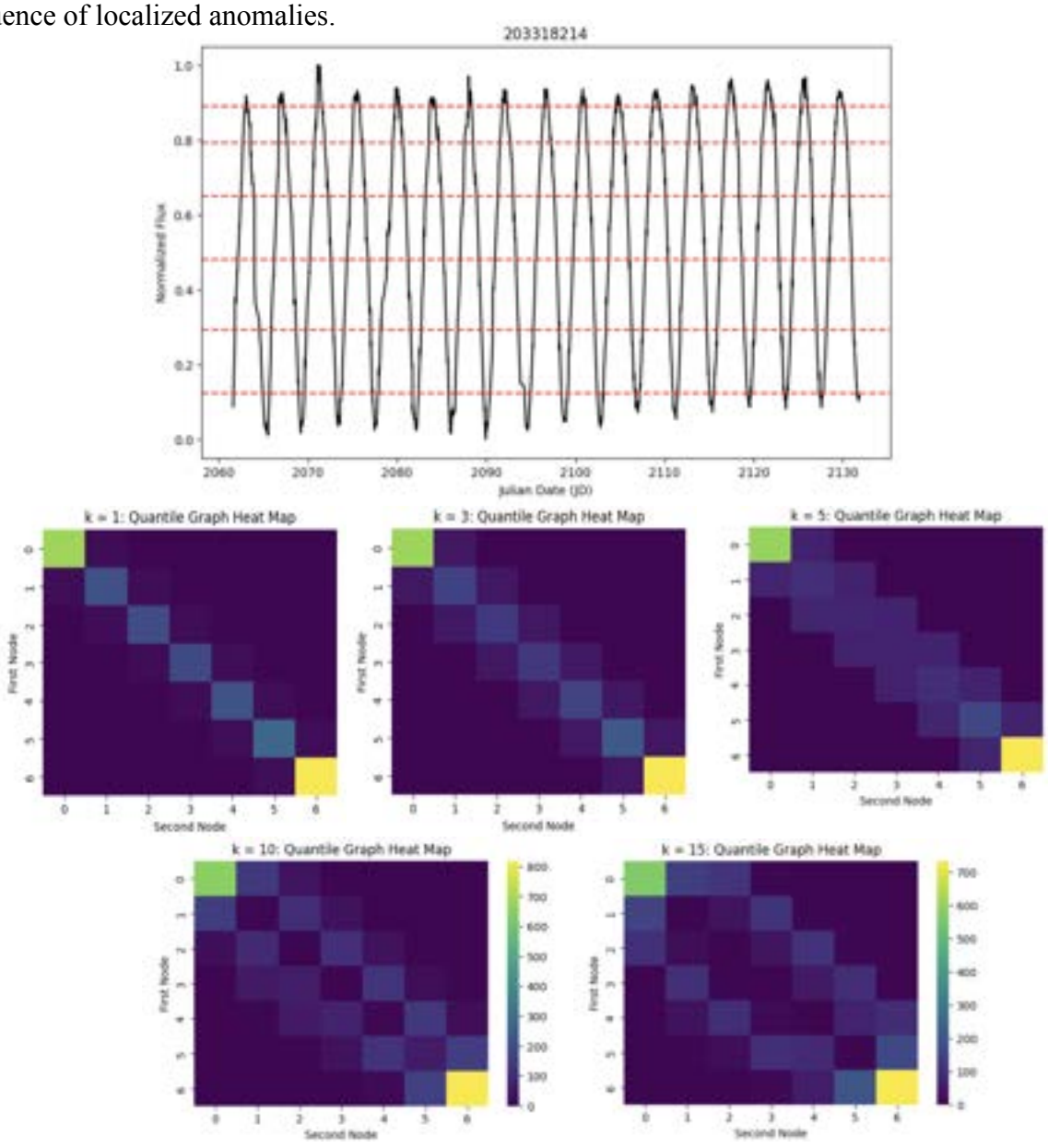


Fig. A1. Quantile graphs are shown with varying k values for K2 light curve with EPIC ID 203318214.

Appendix B

To calculate the quantiles, we explored two distinct approaches: calculating the quantiles across the entire dataset and computing individual quantiles for each light curve. The latter method proved to be highly ineffective, as it obscured the distinctions between light curves in the resulting quantile graphs, rendering the variations nearly indiscernible (see figure B1).

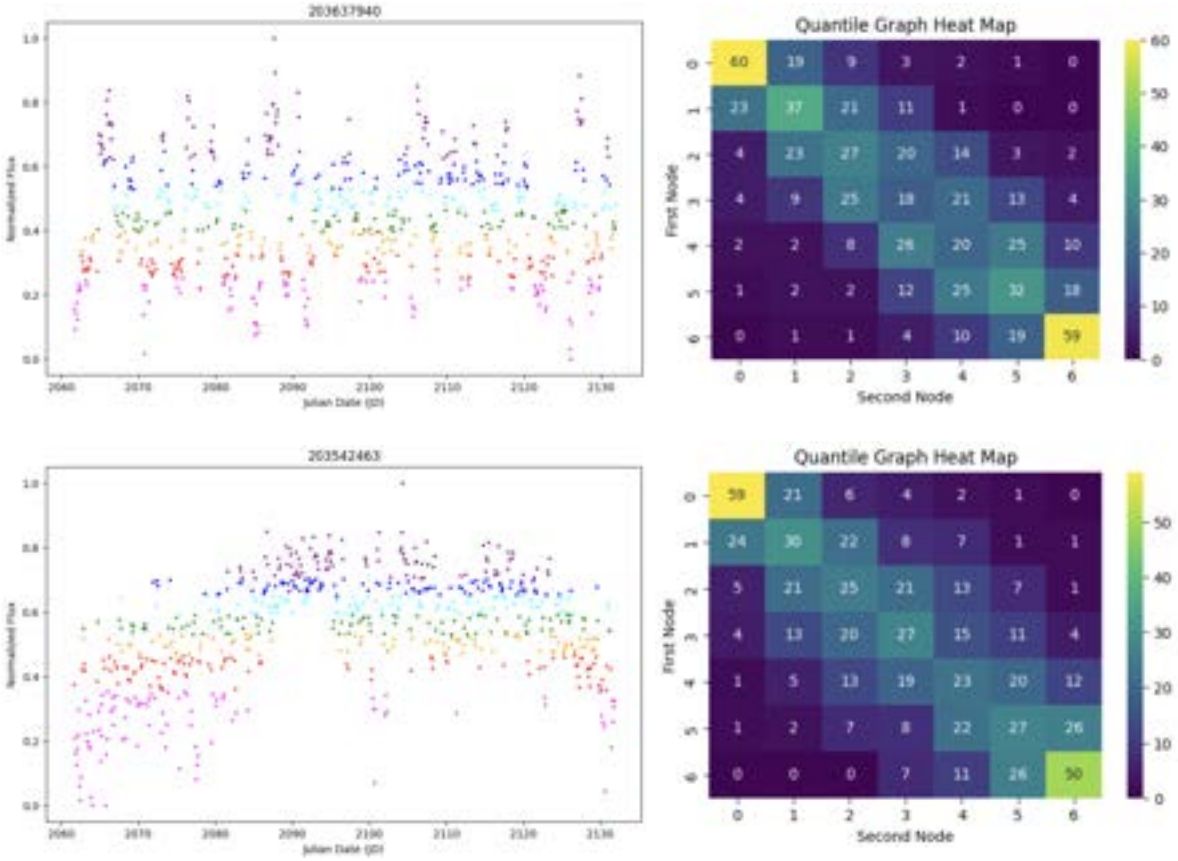


Fig. B1. Quantile graphs with quantiles calculated for each individual light curve. Color coded light curves by quantile on the left, quantile graph heat map on the right.

We also plan to test the optimal number of quantiles to use. Currently, we use 7, as it is a prime number, which minimizes the risk of divisibility by smaller integers, thereby reducing the likelihood of symmetrical patterns that could bias the analysis. Additionally, the choice of 7 strikes a balance between granularity and computational efficiency, making it a favorable option for machine learning applications. However, exploring other values could potentially reveal a more effective choice, enhancing the accuracy and robustness of our model.

Appendix C

Spectral clustering offers 15 key parameters that users can adjust to tailor the algorithm to their specific needs.

num_clusters	Number of clusters to be formed
eigen_solver	Algorithm for eigenvalue decomposition.
n_components	Number of eigenvalues
random_state	Seed used by random number generator
n_init	Number of times k-nearest neighbors algorithm run with different initializations
gamma_kernel	Influence of distance between points in similarity matrix (if radial basis function kernel)
affinity	Method to compute affinity matrix
n_neighbors	Number of neighbors (if k-nearest neighbor used for affinity)
eigen_tol	Stop criterion for eigenvalue decomposition
assign_labels	Strategy for assigning labels in the embedding space
degree	Degree of polynomial (if polynomial kernel)
coef0	Independent term in kernel function (if polynomial or sigmoid kernel)
kernel_params	Pass in extra parameters to the kernel function if applicable
n_jobs	Specify number of CPU cores used for computation
verbose	Prints out logging and diagnostic messages to terminal

Table C1. Spectral clustering parameters and what they affect[10].

The first parameter we adjusted was *num_clusters*, where we experimented with a range from 2 to 25 clusters beyond the default setting of 8. Ultimately, we decided to retain the default of 8 clusters, as it aligns with the 8 recognized variability classes, ensuring that each class is represented distinctly. Additionally, setting the number of clusters to 8 helps maintain a clear

correspondence between the algorithm's output and the established astrophysical classifications, making the results more interpretable and scientifically meaningful.

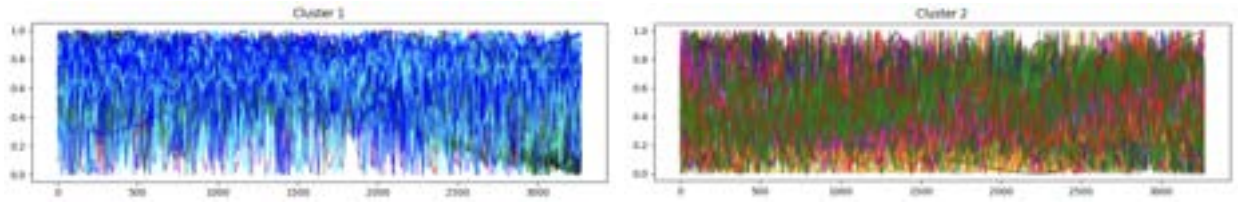


Fig. C1. *num_clusters* = 2

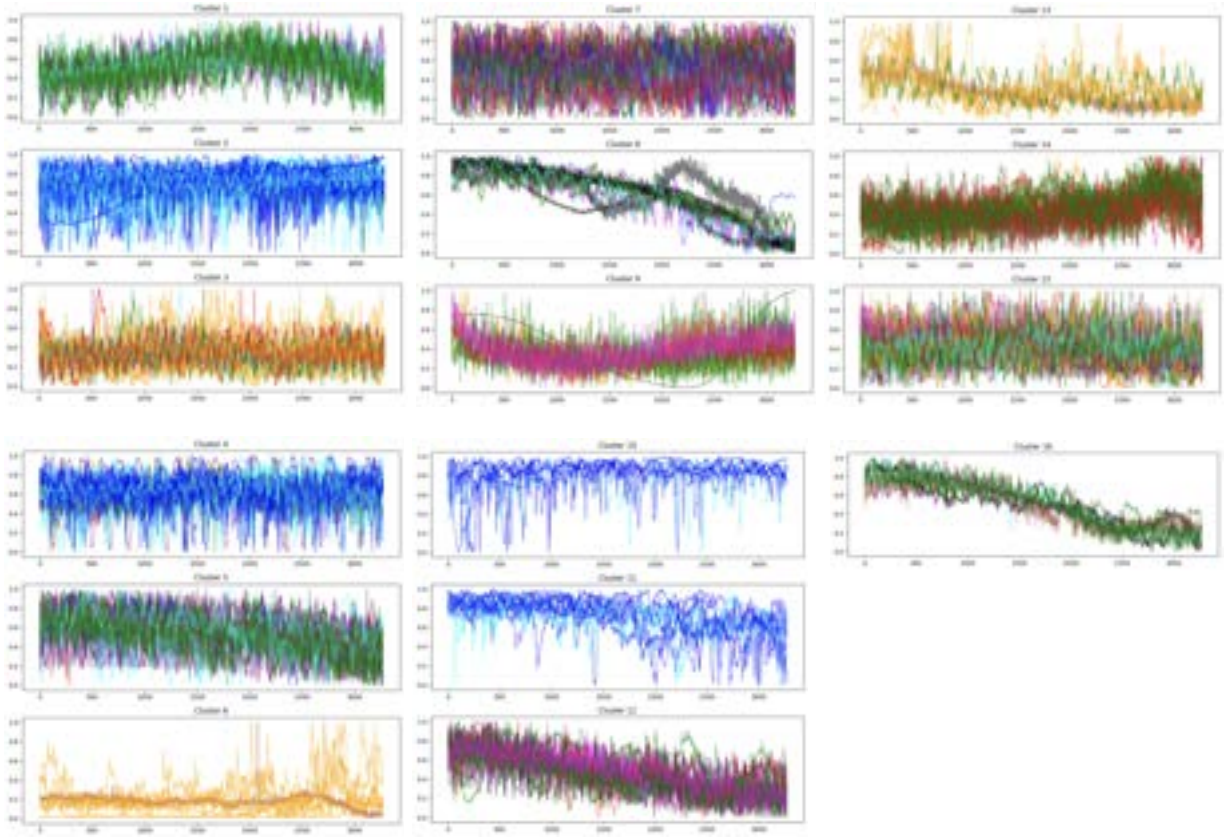


Fig. C2. *num_clusters* = 16

We selected k-nearest neighbors for the *affinity* parameter, as it is particularly well-suited for spectral clustering due to its ability to capture both local and global data structures effectively. This choice required us to determine the number of neighbors, *n_neighbors*, which we set to the default value of 10. This value strikes a balance between capturing local neighborhood information and preserving the global structure of the data, while also being computationally efficient. Similarly, we retained the default *n_init* value of 10 to reduce the risk

of suboptimal solutions, ensuring consistency and quality in the results, particularly given the manageable size of the dataset. By opting for k-nearest neighbors, we determined that the *gamma*, *degree*, and *coef0* parameters were not necessary. Additionally, since there was no need to pass any extra information, we left the kernel_params parameter untouched.

We opted to keep the *eigen_solver* and *n_components* parameters at their default settings, which are arpack and None, respectively, as these are standard choices when the optimal number of components is unknown. We also did not modify *eigen_tol*, typically set to auto, since this parameter is not usually exposed to the user. After computing the eigenvalues, we retained the default *assign_labels* setting of k-means, as the alternative discretize option is more complex to implement and is applicable in fewer cases.

For *random_state*, we used the default value of 42 to ensure reproducibility, a common programming convention that could technically be replaced by any integer, provided consistency is maintained. We also kept *n_jobs* at its default value of 1 CPU core to ensure compatibility and predictability across different systems. Finally, while we enabled *verbose* for debugging purposes during development, we disabled it in the final version to avoid cluttering the terminal output.

Appendix D

We had initial concerns that spectral clustering might not be functioning as intended and that it could be overly influenced by factors such as time scale, Q, M, or other specific values associated with the light curves. If this were the case, it might have been more appropriate to analyze these individual parameters directly. To address this issue, we conducted preliminary tests to ensure the robustness of the spectral clustering algorithm. Specifically, we applied it to a set of randomly generated sine and cosine waves with added noise, arbitrary phase shifts, variations in period, and amplitude difference, each within a reasonable range for machine learning. For our purposes, a "reasonable range" was defined as variations that are likely to occur naturally in astrophysical data without distorting the inherent periodicity or overall behavior of the signal.

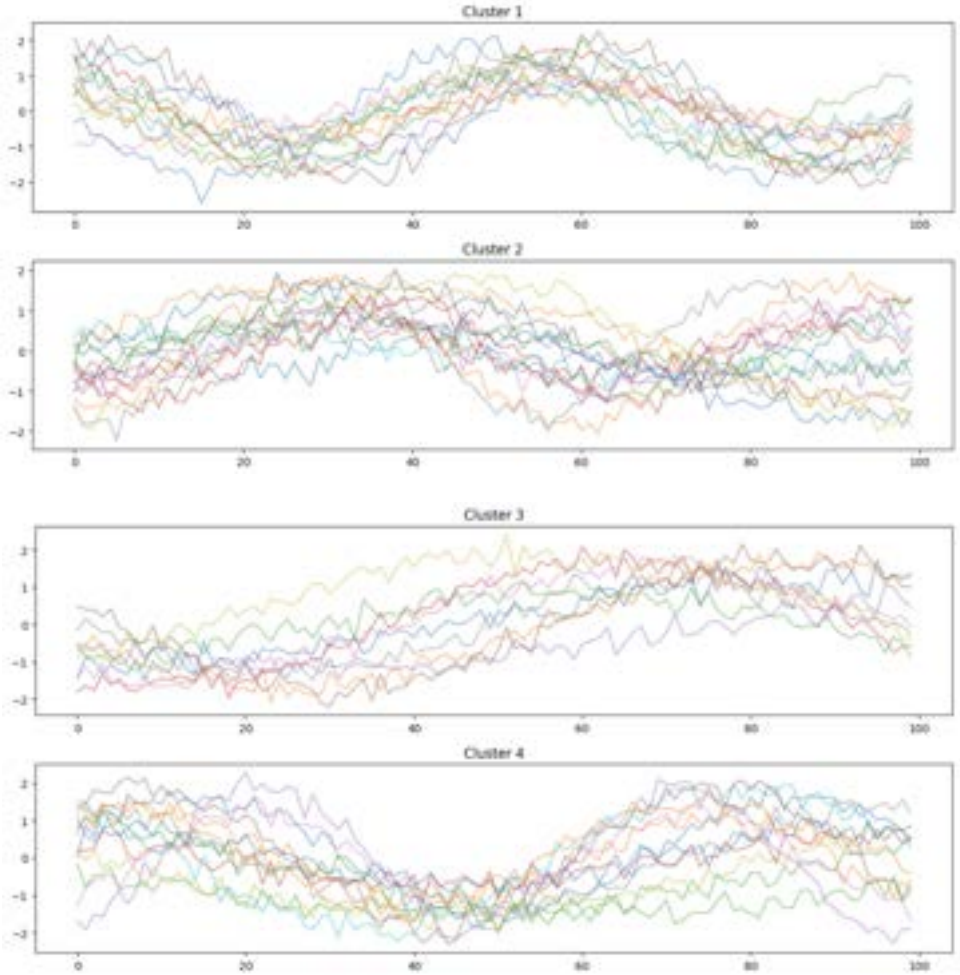


Fig. D1. Four clusters of sine and cosine waves formed with random noise and phase shifts in range [-π, π], period variations in range [0.5, 2.0], and amplitude variations in range [0.5, 2.0].

There were also concerns that the spectral clustering was picking up on the period, Q, or M values rather than the behavior of the light curves. To do so, we plotted the distribution of the initial period, Q, or M values and then applied a transformation to obtain a more uniform distribution and then obtained the quantiles in the same number as the number of clusters so we could ensure there was a mostly equal distribution of color-coded groups across clusters. After plotting the clusters color-coded using these features separately, we noticed that the period and cluster were completely unrelated, but there was a slight correlation between Q or M values and clusters, which makes sense because these features are correlated with the variability classes.

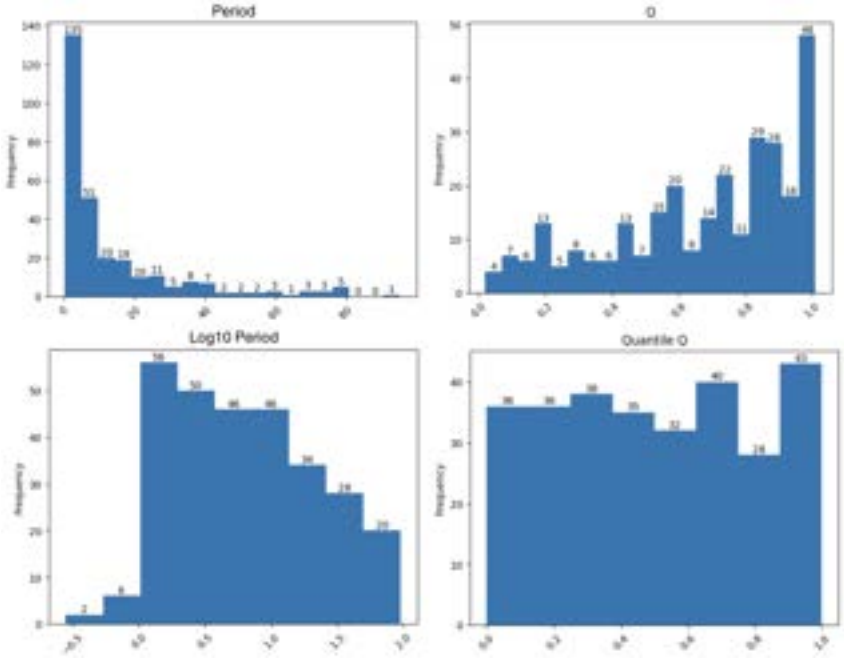


Fig. D2. Histograms of initial and post-processing for period and Q value distribution.

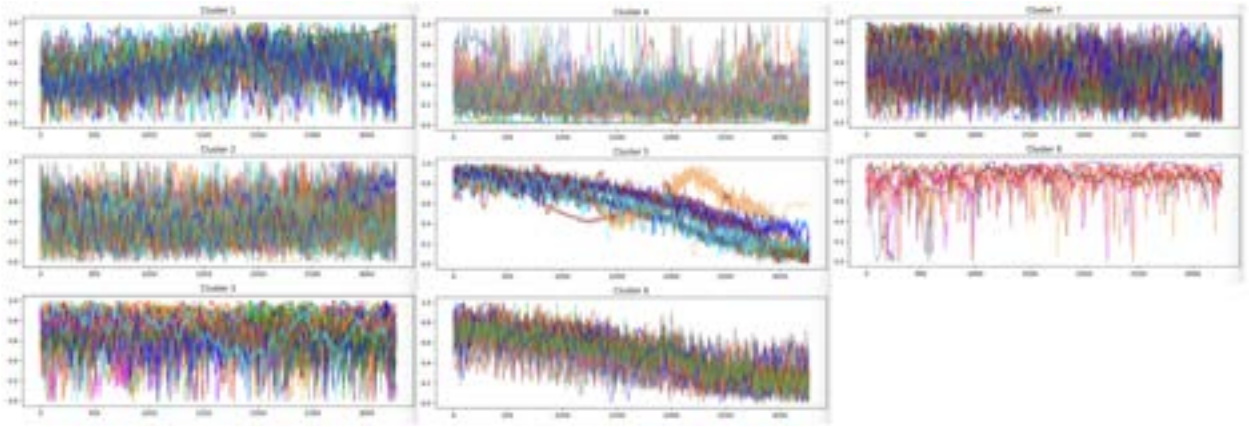


Fig. D3. Clusters color coded by M value quantiles.

To ensure that the program was functioning as intended, we first explored the use of quantile graphs by applying them to a set of generated time series as a preliminary check. The success of these initial tests confirmed the effectiveness of this approach, leading us to proceed with utilizing quantile graphs in our subsequent analysis of light curves.

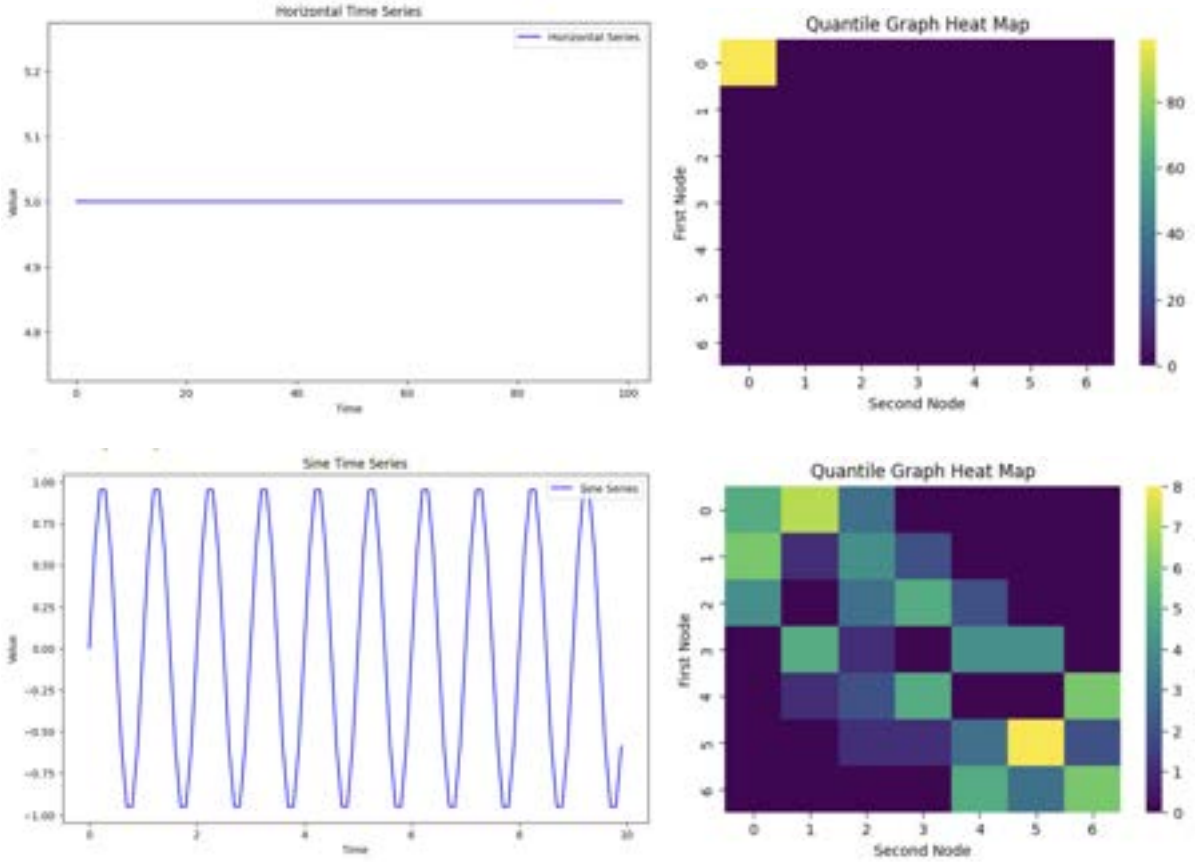


Fig. D4. Sample time series and their corresponding quantile graphs.

An alternative preprocessing method we explored, should quantile graphs prove inadequate, was dynamic time warping (DTW), a technique that aligns time series data by non-linearly stretching or compressing time intervals to minimize the distance between sequences. While DTW is effective in handling phase shifts and varying temporal patterns, we ultimately chose not to use it because it does not account for differences in periods, which is critical for our analysis. Nevertheless, it was an interesting experiment that may have applications in addressing other types of time series challenges.

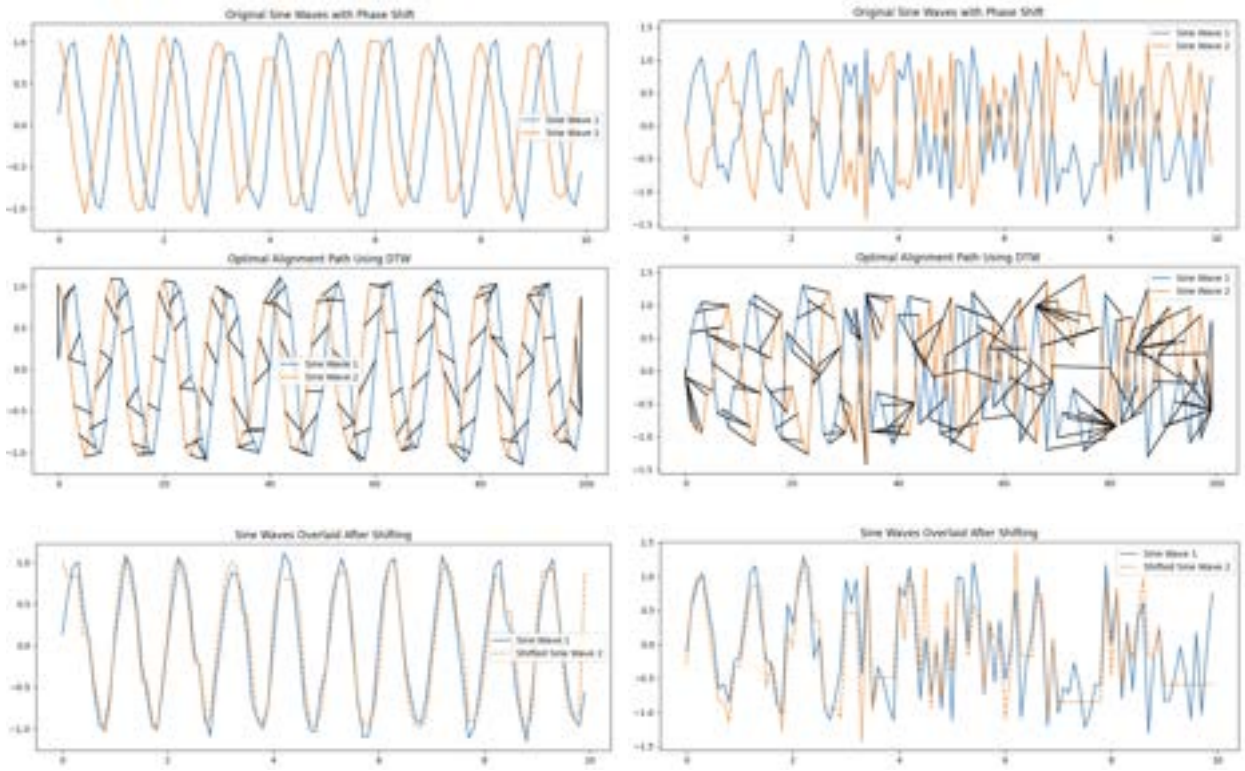


Fig. D5. Sine waves with varying levels of randomness, aligned and overlaid using DTW.

Appendix E

The corresponding code and data can be accessed via the links provided below.

K2 (Upper Scorpius Cluster)	Link
K2 (Taurus Cluster)	Link
TESS (Taurus Cluster)	Link
K2 (Upper Scorpius Cluster) - testing *num_cluster*	Link
K2 (Upper Scorpius Cluster) - testing clustering algorithms	Link
K2 (Upper Scorpius Cluster) - plotting time scale, Q, and M clusters	Link
K2 (Upper Scorpius Cluster) - before quantile graphs introduced	Link
K2 (Upper Scorpius Cluster) - test mean normalization	Link
K2 (Upper Scorpius Cluster) - calculate quantiles on individual light curves	Link
K2 (Upper Scorpius Cluster) - UMAP and PCA	Link
K2 (Upper Scorpius Cluster) - test *k* values for quantile graphs	Link
K2 (Upper Scorpius Cluster) - quantile graphs	Link
K2 and TESS (Taurus Cluster) - side by side plot comparison	Link
Testing Spectral Clustering (varying period, amplitude, random noise)	Link
Testing Quantile Graphs (introduction to DTW)	Link
Testing UMAP	Link

Table E1. Links to Google Colab notebooks.