



Visualization of Repeated Patterns in Multivariate Discrete Sequences

Konstantinos F. Xylogiannopoulos
 Dept. of Computer Science
 University of Calgary
 Calgary, Canada

 <https://orcid.org/0000-0003-2376-898X>

Panagiotis Karampelas
 Dept. of Informatics & Computers
 Hellenic Air Force Academy
 Dekelia, Greece

 <https://orcid.org/0000-0003-1684-7612>

Abstract— The availability and affordability of mobile devices, wearables, sensors, IoT devices and electronic social networks produce big data in the form of complex systems of multivariate discrete sequences such as bio-informatics, natural language processing, social network corpus, etc. or non-discrete time series such as weather data, network traffic, workout data, etc. At the same time, the increased availability of advanced hardware in the form of powerful computers or high performance clusters provide us with the opportunity to analyze the aforementioned datasets that could produce vast amounts of results in diverse forms. One problem that has recently got focus is the one of discovering all repeated patterns in multivariate sequences. Novel algorithms have appeared such as ARPAD that address the specific problem however there are still no appropriate visualization methods to represent the complex results of the algorithm. In this paper, we attempt to create a visualization method that presents the common repeated patterns in multivariate discrete sequences. The visualization algorithm has been applied in a dataset of different text sequences of varying length and the results are presented in two novel type of plots, the Pattern Positional Alignment (PaPA) plot and the Stacked PaPA plot.

Keywords— Visualization, repeated patterns, complex information representation, ARPAD

I. INTRODUCTION

The recent years data analytics have attracted the focus of the academic and research community as well as business professionals. This is mainly attributed to the uninterrupted sequences of data produced by the contemporary devices that everyone has. Smart phones, tablets and wearables are some of the devices that are continuously connected to the Internet feeding user data flows to the corresponding online social networking platforms. At the same time, IoT devices and sensors collect data of various types either observing the weather, the network traffic or even the pedestrians crossing an area in a city. People seeking to understand human behavior, physical phenomena or to extract business intelligence are keen on analyzing those data by employing diverse scientific techniques.

A common problem in data analytics that has recently got focus is the discovery of all repeated patterns in multivariate sequences. A novel algorithm that is able to address the specific problem is ARPAD [1, 17] which detects all repeated patterns found in multivariate discrete or non-discrete sequences that have been stored in a LERP-RSA data structure [1, 2, 17]. The algorithm is able to identify the sequence, the position of the common pattern and the pattern itself. In complex sequences, the reported results are usually from thousands to billions, depending on the application domain and sequence size. For example, in network traffic analysis [3], the algorithm is able to detect all the Internet Protocol (IP)

addresses prefixes and based on the specific analysis can notify the network administrator whether a Distributed Denial of Service attack is underway. In another application domain, ARPAD can detect all the repeated textual patterns in a corpus of several documents revealing potential cases of plagiarism [4]. In weather analytics, the algorithm has been used to detect similar patterns in temperature and geopotential height for the isobaric surface of 500 mbar for different locations [5] producing the latitude and longitude of the same pattern found at a specific time. Furthermore, the algorithm has been applied in the analysis of a number of programming assignments in order to detect similar code that has been used by the students reporting a great number of similarities [6]. In all the aforementioned cases, the algorithm reports the pattern detected, it's length, the number of occurrences, the positions that the pattern exists and the identification number of the sequence in which was found (Table I).

TABLE I. ARPAD RESULTS

Pattern	L.	Occ.	Positions	Ref
647227219946156061860 22	23	2	2806926755,486503740532	[17]
:=ord()+[]=chr()print(.end =)=+	30	3	20;69,35;28,38;60	[6]
bbbbbbbc	8	31	0;1159,0;1415,0;1427,0;1439,0;15 63,0;1655,0;1679,0;1687,0;171,0; 1711,0;2683,0;275,0;315,0;35,0;5 9,0;87,1;107,1;1431,1;1500,1;155 1,1;1580,1;1599,1;1752,1;1791,1; 1871,1;199,1;233,1;2619,1;2666,1 ;2689,1;336	[5]
GCCTGTAGTCCCAGCT ACTCGGGAGGCTGAG GCAGGAGAATGGCGT GAAC	50	6082	...	[16]
74.54.182.xxx	9	655302	...	[3]

The textual report of the algorithm, even if it is very useful since it detects all the repeated patterns in all different sequences which no other algorithm can achieve to the best of our knowledge, the analysis of the results is only attainable using textual queries. This means that the analyst should know beforehand what exactly he or she is looking for and thus interesting but unknown or unexpected results may be missed. Our current work attempts to solve the specific problem by proposing a visualization method that is able to present the common repeated patterns in multivariate discrete sequences in a graphical manner. The visualization method is able to produce two different diagrams with the relevant information: (a) a graphical representation of each sequence and all the patterns found aligned with the length of the sequence based on their absolute position; and (b) a graphical representation of each sequence with all the common patterns stacked without their absolute position in order to display proportionally the portion of each sequence that can be found in other sequences too. The specific method is the first one

implemented in order to visualize the complex results produced by ARPaD algorithm and can provide the analyst with an exact representation of the positions of all the repeated patterns detected by the algorithm or the common portions among different discrete sequences analyzed by the algorithm, something that was not feasible otherwise.

The paper is structured as follows: Section II analyzes similar work in the area of visualization of complex data in general since there are no other similar visualization techniques that achieve the same outcome as the one proposed in this paper. In Section III, the proposed visualization methodology is described, presenting the two different approaches. Section IV illustrates the results of the proposed visualization technique applying the method to a complex dataset and discusses the hidden clues discovered by applying the specific methodology. The final Section concludes the paper summarizing the most important aspects of the work presented and sets the basis for the future work in the area.

II. RELATED WORK

Data visualization is a very effective communication channel in data analytics. The value of visual analytics has been recognized by the researchers very early and progressively has been used in more complex techniques such as cluster analysis of two dimensional data [7] or text mining in which large collection of text is analyzed [8]. In both cases, as the results of the analysis are huge, it is not easy to extract useful conclusions using computational methods. In [7] the authors propose three different visualization techniques based on a sliding window in order to visualize patterns identified in nucleotides while in [8] a tool is presented that is able to present the repeated n-grams among different documents that are analyzed. The first methods are only focused in a specific dataset and find application in DNA analysis while the second work focuses on textual analysis similar to our approach, however, addressing a very narrow problem that of visualization of n-grams in text documents while our visualization approach is more generic and can be applied in multivariate discrete sequences and not only text documents.

Other techniques, such as the one presented in [9] proposes the use of different visualization techniques to represent repetitive patterns found in large password datasets. The specific work proposes character frequency map, short repeat heatmap, long repeat parallel coordinates and repeat word cloud. It is argued that the specific techniques can work together and provide the analyst with the necessary information for the specific dataset. Since the dataset usually comprises short phrases, i.e., passwords that have a specific meaning which is understandable to human beings, the proposed techniques can serve their purpose and be useful in the analysis of the patterns detected. In a generic case in which the pattern does not have a specific meaning, i.e., in weather analytics, the application of the proposed techniques is not so efficient for the analysis.

Another work in visualizing very large and complex data has been proposed in [10] in which a method that is based on the organization of the available data in groups and subsequently arrange the groups in a suitable manner and produce a global pattern. The specific technique has been applied in a financial dataset which is a time series applying specific order suitable for the specific series. However, while the aforementioned technique can be effective in the representation of very large amount of financial data, in cases

where the relative position of patterns inside a dataset is required, the specific technique cannot be used.

To solve the problem of visualizing very large datasets, in [12] a similar approach is proposed. Data summarization and segmentation is proposed in order to address the large amount of the available data adopting a data mapping approach in which portion of the data can be shown and the analyst can visually browse in different areas of the dataset and zoom in or zoom out when something interesting is found. While the technique may address the issue of visualizing very large datasets, it cannot address the representation of repeated patterns detected in the dataset.

A similar approach that is based on data reduction and summarization for pattern visualization is presented in [12]. While the specific technique attempts to handle a very similar problem that our approach addresses, the implementation of the method has been done in a very small and rather simple dataset and thus it is not clear how the specific method handles larger datasets. As it is stated, the method visualizes only "consecutive" patterns and not patterns found in different places in the dataset and thus losing valuable information. Our method is able to visualize all the patterns detected of any length in all the different sequences even if there is overlapping.

Nataraj et al in [13] adopts a traditional visualization technique in order to present similarities between computer programs and more specifically malware. The computer program is visualized as an image in which its different parts are created based on the code structure of the program. In order to compare the malware and find the similarities the images are compared and a confusion matrix is presented. While the specific technique can effectively represent the classification of the malware programs, does not adequately represents the common patterns between the computer programs.

Livieri et al in [14] have analyzed a large dataset with the source of computer programs approximately of 400 million lines of code and attempted to the most frequently used code. In the experiment, eighty PCs were used and after a two day analysis, the desired results were produced. Then the results were visualized using a scatter plot presenting the inter-project code detected in the dataset. The diagram has been enriched with information related to the category of code identified and thus can be easily understood. However, in cases where there is no such categorization or the identified patterns are unknown or do not have a specific meaning then the specific method is not effective since it will not be possible to present the label of the common patterns. Moreover, other important information such as the length of the computer program or the position of the pattern detected is not possible to be visualized in the specific approach. Our proposed method addresses all these shortcomings in an efficient and effective way.

Finally, in [15] the authors attempt to categorize all the visualization methods used to compare data. They identify that there mainly three basic categories in which the methods can be distinguished. In the first category, the visualization methods use juxtaposition to present the data compared. In the second category, the methods use superposition to represent the data in a common context and finally in the third category the methods attempt to visualize the relationships among the data represented. They have also found that there are hybrid methods combining two of the aforementioned categories to

present the common patterns. Our approach based on the specific categorization falls in the hybrid methods in which all the specific techniques have been combined in order to better visualize the common repeated patterns found in the multivariate discrete sequences as it will be presented in the following section.

III. PATTERN POSITIONAL ALIGNMENT

In previous sections, it has been discussed the problem of visual representation of repeated patterns in very large datasets. Mainly, this problem occurs because ARPaD algorithm can detect all repeated patterns that exist not just in a single sequence but most importantly in a multivariate system of sequences. For such kind of multivariate systems, it is very important to identify patterns that fall in several different categories. The first category is the identification of patterns that exist in a single sequence. The generalization of this standard problem in multivariate systems is the detection of repeated patterns in a sequence for every sequence. The second category, which is an even broader generalization of the first category, is detecting repeated patterns that exist in multiple sequences. This category basically connects already discovered patterns among sequences. The third category, which in several cases is the most important and difficult to identify, is detecting repeated patterns that exist only among different sequences, i.e., they do not repeat in a single sequence but only among two or more. This is the most difficult category to be identified since a per sequence analysis is not possible; the full multivariate system of sequences must be analyzed as a whole in order to detect those patterns.

In order to perform multivariate repeated pattern detection, there are several phases of pre-processing that have to be executed. First, we need to clean our dataset and prepare it for the next phases of the analysis. This step is important since any irrelevant data values could lead to erroneous representation of the sequences and possible failure of the algorithm to detect patterns because of them. The next phase is the discretization of sequences and transformation to strings. This step is fundamental for sequences having continuous values such as weather time series. Sequences of this type have to be transformed to discrete strings using a finite alphabet. Moreover, this phase can also be used for sequences that are already discretized and where a use of another alphabet can significantly help to reduce the size of the sequences and improve performance, e.g., sequences of proteins where the three-letters representation, such as *ALA* for Alanine, can be transformed to the one letter alphabet of proteins, *A* for the abovementioned example. The third phase is the creation of the multivariate LERP-RSA data structure where a complex table is created holding information about suffix strings, their positions per sequence and the sequence that they exist. In the next, fourth phase, the ARPaD algorithm runs on the LERP-RSA data structure and identifies all repeated patterns that exist for every category it has been mentioned at the beginning of the section. Finally, several meta-analyses can be executed based on the desired outcome we want to evaluate. For example, in several cases we might not be interested about overlapping patterns rather than we care only about the longest, distinct, patterns we can observe. This is common for text mining cases where we could care about the identification of partial or full sentences than words [4] or very long patterns that can help the clustering of sequences [6].

The results of ARPaD algorithm usually are vast in comparison to the size of the dataset, yet, in most of the cases we care to visually represent only partial, yet important, results. Still, this can be extremely difficult trying to visually represent on a 2-dimension graph a complex multivariate system of sequences and their repeated patterns. For this reason, the following Pattern Positional Alignment (PaPA) algorithm will be presented that helps to organize the results and visually represent them in the best possible way.

- 1) *Sort patterns found by ARPaD in descending order by their size*
- 2) *For every pattern, identify the sequences where pattern exists*
 - A. *Give to pattern the level index 1*
 - B. *For every sequence the new pattern exists check if it is the first one detected*
 - I. *If true then repeat from step (2)*
 - II. *Otherwise for every pattern found and has the same level index*
 - a. *Check if the position of the new pattern is between the position of the pattern already found and its ending position OR the ending position of the new pattern is between the position of the pattern already found and its ending position*
 - i. *If true for any pattern already found then increase level index by one and repeat from step (ii)*
 - ii. *Otherwise go to step (2)*

The algorithm terminates because it is based on three nested loops. The outer loop (2) runs over all patterns found. The first level inner loop (B) runs over the sequences that a pattern has been identifying to exist. The second level inner loop (II) runs over the patterns that have already being classified for every sequence of the outer loop (2). When the patterns classified finish, it will continue on the outer loop (2) until all patterns are scanned and classified.

The worst-case complexity of the algorithm is when having all existing patterns in only two sequences. In this case, the algorithm in step (a) has to check a new pattern against all patterns already classified. Therefore, the time complexity is $O(p_t^2)$ where p_t is the number of patterns found by ARPaD. The average case, which is also the best case, is if we assume that patterns are equidistributed over all sequences. Now, every sequence will have approximately $p_s = p_t/n$ patterns, where n is the number of sequences and p_s is the number of patterns found per sequence. Thus, the time complexity will be $O(np_s^2) = O(n(p_t/n)^2) = O(p_t^2/n)$.

Finally, the algorithm is correct because for the new pattern, it firstly assigns the level index 1 (A) and then checks if it is the first found for the particular sequence (B). If it is, then the pattern keeps the level index 1 (I) otherwise it checks if it overlaps with a pattern already found and has the same level index (a). If not, it maintains the level index for the new pattern (ii), otherwise it increases the new pattern level index by 1 and repeats the process until either finds a level that it does not overlap with any other previously found pattern or moves it to a new, higher level where it is the first pattern of

that level (i). Therefore, the new pattern is either placed in a level without overlapping with any other pattern or it is placed on a new level where no other pattern exists. Moreover, overlapping checks cannot fail since the patterns are sorted by size from longer to shorter. Thus, it is impossible for a new pattern to be longer than a previously found pattern and have starting and ending points simultaneously outside the boundaries of the previously found pattern. Therefore, the algorithm is correct.

We can observe that PaPA algorithm is analogous to scheduling problem algorithms where instead of time alignment we need to perform spatial alignment. Yet, there are significant differences, such as the multilevel alignment of the patterns in comparison to the restricted parallel execution of processes in the scheduling algorithms.

IV. DISCUSSION

The aforementioned PaPA algorithm helps us classify the patterns per sequence by assigning them a level index. This level index can be used to create two different type of visualizations. The first diagram is the PaPA plot while the second is the Stacked PaPA plot. The standard PaPA plot, has a horizontal grey bar for each sequence illustrating its length. On top of this grey bar, each pattern that has been found and belongs to the sequence is placed at its position in the sequence and at the appropriate level that has been classified by the PaPA algorithm. The height of the patterns inside the sequence bar is always an aliquot part of the height of the sequence bar over the highest level index observed by the PaPA algorithm. This allows the best possible fitting for the patterns and has the best possible visualization results. For easier identification of the patterns on the plot, a different color and motif has been assigned to each pattern. For better results, the number of colors and motifs are both prime numbers in order to guarantee the highest possible combination of colors and motifs before they start to appear again, in case that the patterns number is higher than the number of colors and motifs.

In the second type of plot, the Stacked PaPA plot, the sequences are again represented with a grey horizontal bar. However, the patterns now are plotted one next to other following the order of appearance based on the position in the sequence. Again, each pattern has different color and motif and, moreover, on top of it, it is printed a triplet of numbers presenting the pattern index, its position and length. The height of the patterns now is a fraction smaller than the height of the grey bar since patterns are not plotted in levels. More importantly, now the total length of the stacked patterns can be significantly longer than sequence length depending on the number of patterns in a sequence and their respective lengths. This give us an important measure of how much the patterns overlap with the sequence length, which can be observed by the grey bar.

For our examples, we have used a subset from the dataset created in [6]. The dataset contains the delivered programming source code produced by 23 students working in groups in order to implement a specific algorithm. Each student submitted its own version which had several similarities with the other students in the group and overall, since all of them had to implement the same algorithm. The dataset has been processed and analyzed using the steps described in the previous section, while several metadata analyses have been performed on the results of ARPaD. For each one of the

metadata analyses, several types of visual results can be illustrated. For example, the longest patterns found have length 88. In “Fig. 1-2” all non-overlapping patterns with length between 40 and 100 have been presented. The first longest pattern found with length 88 belongs to sequences 13 and 22 and appears on both at position 0 with light green color and vertical lines motif. The second longest pattern with length 88 belongs to sequences 2 and 10 and appears at position 0 while it is visualized with brown color and forward diagonal lines. However, sequences 2 and 10 have one more common pattern with sequence 4, with index 4 at position 30 and length 64 (dark blue with vertical lines motif). Moreover, sequence 2 and 4 have a longer pattern between them with length 80 at positions 24 and 30 respectively and color cyan with forward diagonal lines motif. Because of these extra patterns for sequence 2 with very long length, the plotting on top of each other is important in order to be properly visualized (Fig. 1). Moreover, on “Fig. 2” we can observe that sequence 2, with regard to its length, it is significantly more overlapped by the patterns found in comparison to sequences 4 and 10, thus, revealing a higher correlation.

Additionally, two more different ARPaD meta-analyses with different pattern length thresholds have been presented here and visualized accordingly. First, identifying patterns with length greater than or equal to 40 and less than or equal to 50 (Fig. 3-4). Secondly, all patterns having length exactly 40 (Fig. 5-6). In all cases the patterns visualized are non-overlapping and this explains, for example, why sequence 2 presents significantly more patterns with lengths between 40 and 50 than between 40 and 100.

V. CONCLUSIONS

The current paper proposes a novel visualization methodology for repeated patterns in multivariate sequences. The problem of repeated patterns detection has been recently proven of high importance because of the multiple domain applications. So far only ARPaD algorithm can address the specific problem, yet, until now no appropriate visualization method could represent the complex and vast results of the algorithm.

The need for visualizing the complex information that is produced by the ARPaD algorithm, motivated us to attempt to develop a visualization method to achieve this purpose. For this reason, PaPA algorithm was invented, which can align the positions of the common repeated patterns in multivariate discrete sequences in the best possible way revealing valuable knowledge. The outcome of the visualization algorithm allows us to create two novel type of plots for illustrating repeated patterns, the Pattern Positional Alignment (PaPA) plot and the Stacked PaPA plot.

Moreover, several paradigms on different ARPaD results meta-analyses are presented in this paper, to prove the effectiveness of the visualization technique in a number of patterns found in multivariate discrete sequences. In addition, the smart use of colors and motifs, using prime size sets for each one, allows a more compact visualization for easier identification of the common patterns presented in each plot. Where possible, extra information is presented on each plot to further assist knowledge identification. Finally, it is in our intention to further enhance the PaPA plots and attempt to create additional visualizations to better illustrate the knowledge retrieved from the ARPaD results per application domain such as in weather analytics, bioinformatics, etc.

Common Patterns with Length [40, 100]

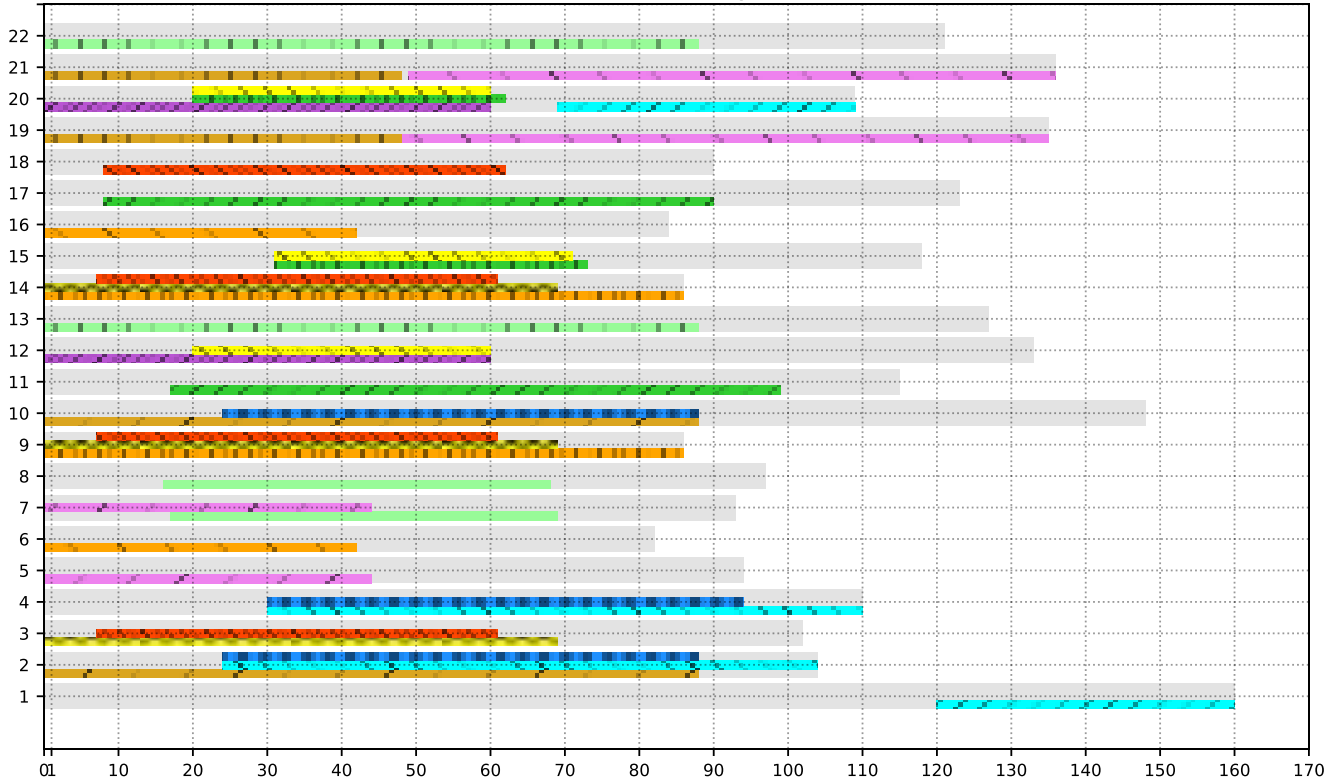


Fig. 1 PaPA plot for common patterns with length [40, 100]

Common Patterns with Length [40, 100]

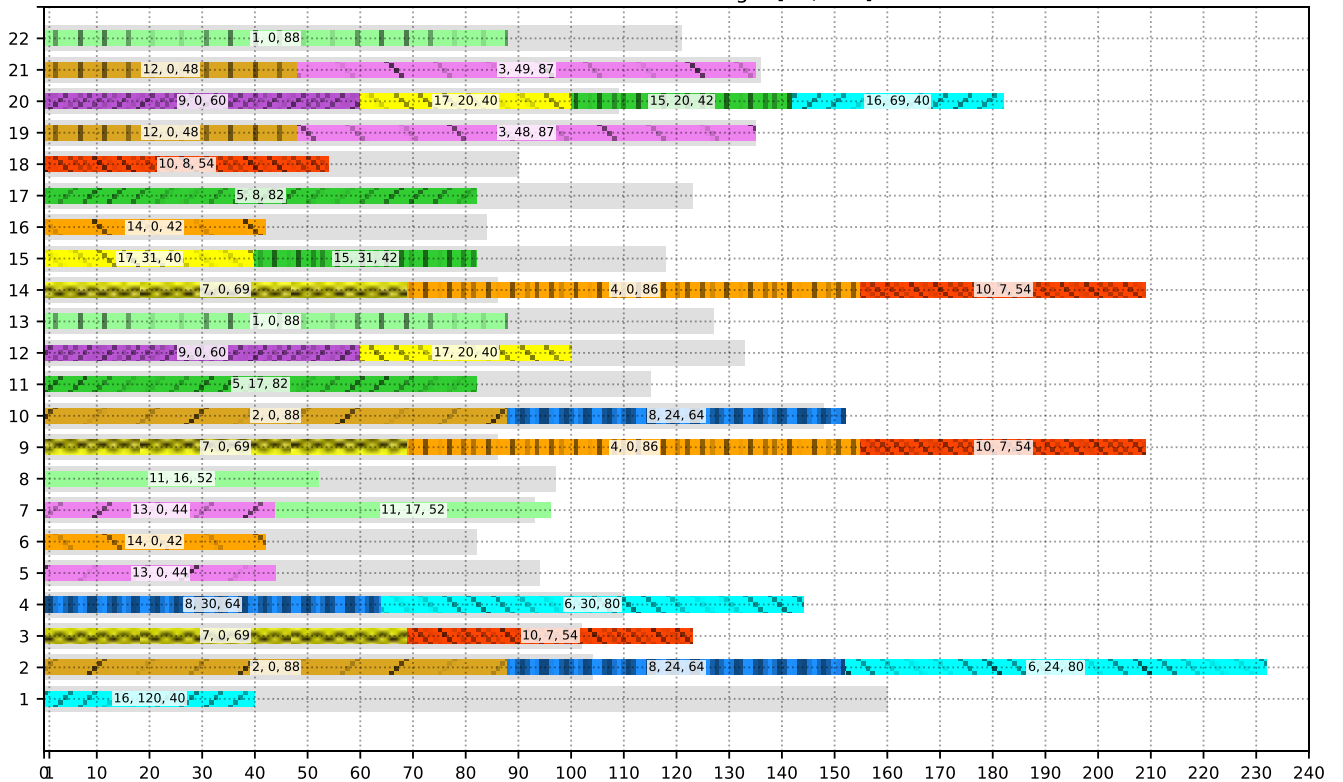


Fig. 2 Stacked PaPA plot for common patterns with length [40, 100]

Common Patterns with Length [40, 50]

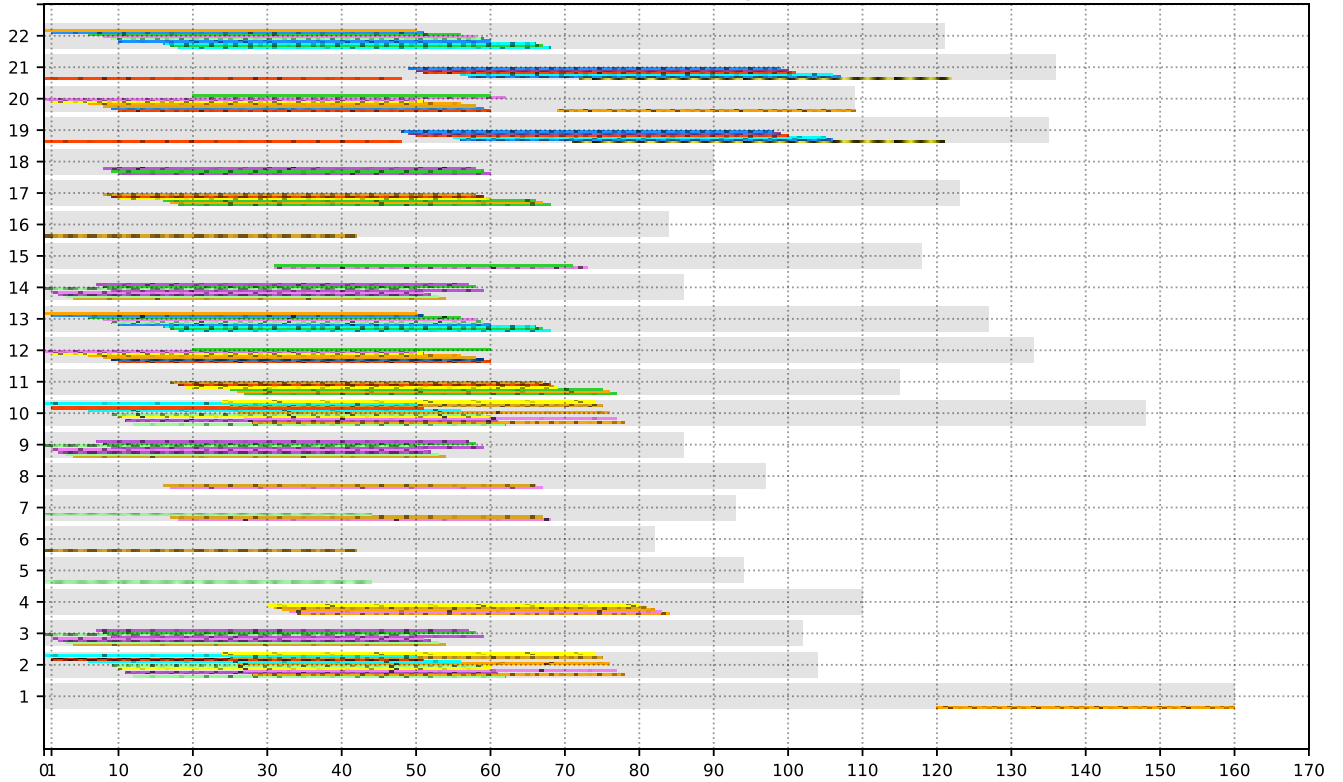


Fig. 3 PaPA plot for common patterns with length [40, 50]

Common Patterns with Length [40, 50]

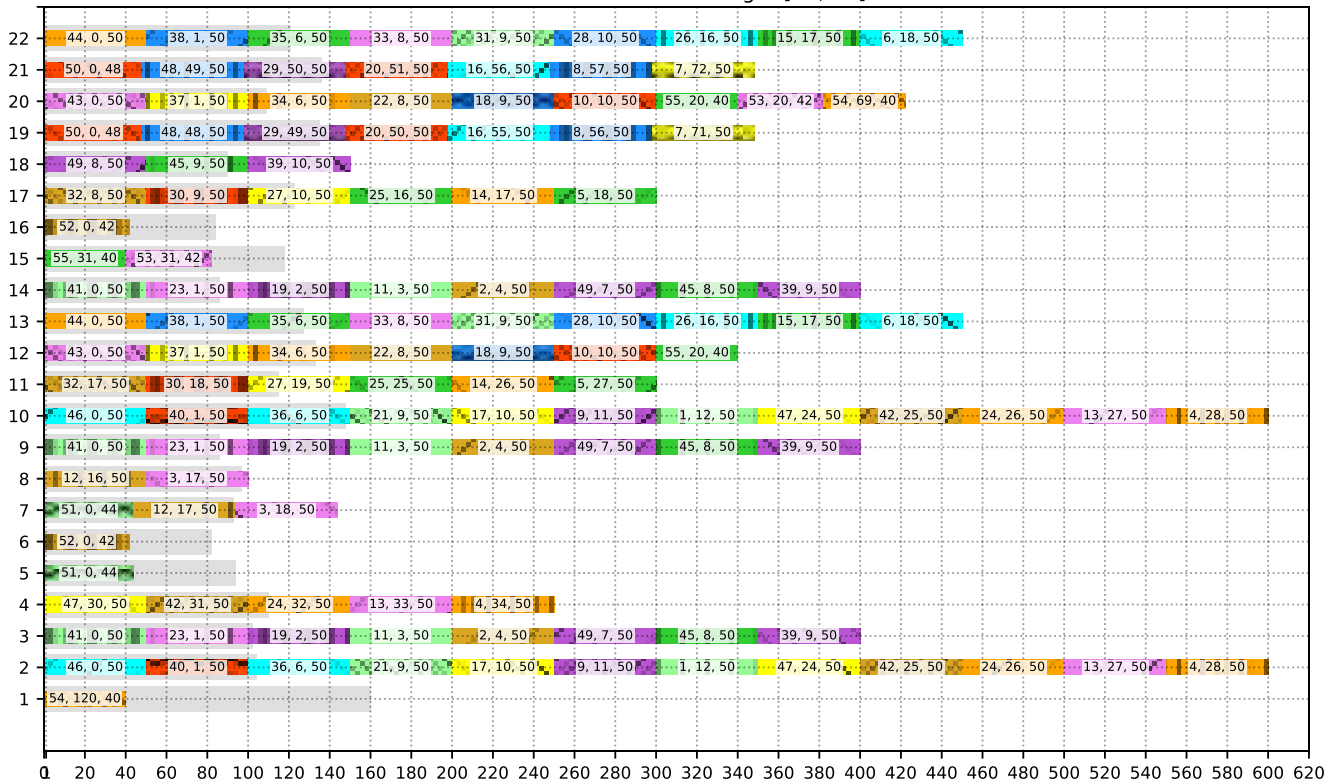


Fig. 4 Stacked PaPA plot for common patterns with length [40, 50]

Common Patterns with Length [40, 40]

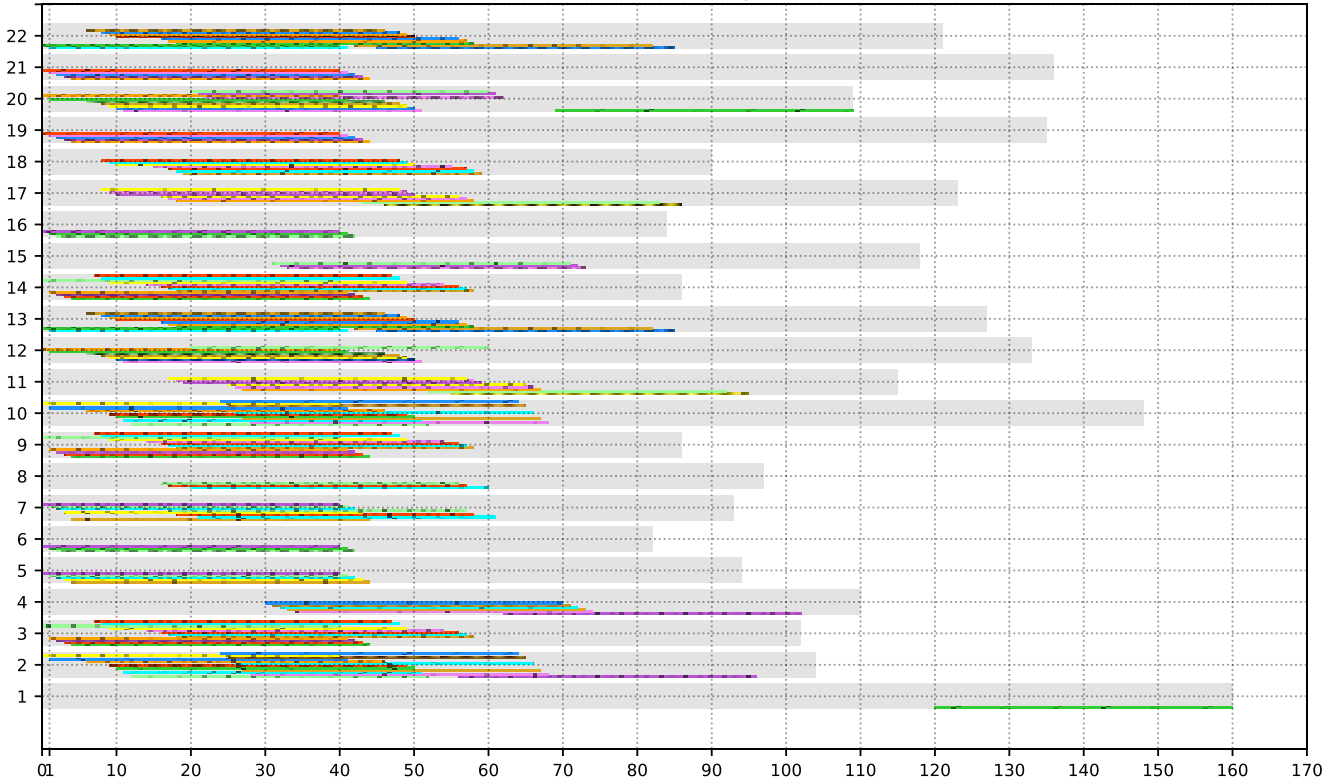


Fig. 5 PaPA plot for common patterns with length 40

Common Patterns with Length [40, 40]



Fig. 6 Stacked PaPA plot for common patterns with length 40

REFERENCES

- [1] Xylogiannopoulos, K.F., Karamelas, P., Alhajj, R. "Repeated patterns detection in big data using classification and parallelism on LERP reduced suffix arrays" *Appl. Intell.* 45(3), 2016, pp. 567– 561
- [2] Xylogiannopoulos, K.F., Karamelas, P., Alhajj, R. "Analyzing very large time series using suffix arrays" *Appl. Intell.* 41(3), 2014, pp.941– 955
- [3] Xylogiannopoulos, K. F., Karamelas, P., & Alhajj, R. (2016). Real time early warning DDoS attack detection. In *Proceedings of the 11th International Conference on Cyber Warfare and Security*,(2016) (pp. 344-351)
- [4] Xylogiannopoulos, K., Karamelas, P., & Alhajj, R. (2018, August). Text mining for plagiarism detection: multivariate pattern detection for recognition of text similarities. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 938-945). IEEE.
- [5] Xylogiannopoulos, K., Karamelas, P., & Alhajj, R. (2019, August). Multivariate motif detection in local weather big data. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 749-756). IEEE.
- [6] Xylogiannopoulos, K., & Karamelas, P. (2020, December). Identifying Social Networks of Programmers using Text Mining for Code Similarity Detection. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* IEEE.
- [7] Mylläri, A., Salakoski, T., & Pasechnik, A. (2005). On the visualization of the DNA sequence and its nucleotide content. *ACM SIGSAM Bulletin*, 39(4), 131-135.
- [8] Don, A., Zheleva, E., Gregory, M., Tarkan, S., Auvil, L., Clement, T., Shneiderman, B. & Plaisant, C. (2007, November). Discovering interesting usage patterns in text collections: integrating text mining with visualization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (pp. 213-222).
- [9] Yu, X., & Liao, Q. (2016). User password repetitive patterns analysis and visualization. *Information & Computer Security*.
- [10] Keim, D. A., Kriegel, H. P., & Ankerst, M. (1995, October). Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings Visualization'95* (pp. 279-286). IEEE
- [11] Nykänen, O. (2013, October). Datamap Visualization Technique for Interactively Visualizing Large Datasets. In *Proceedings of International Conference on Making Sense of Converging Media* (pp. 52-58).
- [12] Knüpfer, A., Voigt, B., Nagel, W. E., & Mix, H. (2006, June). Visualization of repetitive patterns in event traces. In *International Workshop on Applied Parallel Computing* (pp. 430-439). Springer, Berlin, Heidelberg.
- [13] Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security* (pp. 1-7).
- [14] Livieri, S., Higo, Y., Matushita, M., & Inoue, K. (2007, May). Very-large scale code clone analysis and visualization of open source programs using distributed CCFinder: D-CCFinder. In *29th International Conference on Software Engineering (ICSE'07)* (pp. 106-115). IEEE.
- [15] Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. D., & Roberts, J. C. (2011). Visual comparison for information visualization. *Information Visualization*, 10(4), 289-309.
- [16] Xylogiannopoulos, K. F., (2019) "Exhaustive Exact String Matching: The Analysis of the Full Human Genome." *ASONAM 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining – Foundations and Applications of Big Data Analytics* (Vancouver, BC, Canada), pp. 801–808. Association for Computing Machinery, New York, NY, USA, DOI: 10.1145/3341161.3343517
- [17] Xylogiannopoulos, K. F. "Data structures, algorithms and applications for big data analytics: single, multiple and all repeated patterns detection in discrete sequences." PhD thesis, University of Calgary, 2017