Iraqi Journal of Science, 2026, Vol. 67, No. 3, pp: 1715-1726
DOI: 10.24996/ijs.2026.67.3.34

ISSN: 0067-2904

TSSF: Enhanced Phylogenetic Inference through Optimized Feature Selection and Computational Efficiency Analysis

Osama A. Salman*, Gábor Hosszú
Department of Electron Devices, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Budapest, Hungary

Received: 29/9/2024 Accepted: 3/ 3/2025 Published: 30/3/2026

Abstract

In this paper, we introduce an improved feature selection algorithm TSSF designed to enhance computational efficiency in phylogenetic tree construction by evaluating feature subsets and identifying those that contribute positively or negatively to tree topology. Building on the symbolic aspects of pattern systems, which encompass symbols (data), syntax (relations between data points), and layout rules, our approach systematically assesses the effectiveness of feature subsets. This is illustrated using essential phylogenetic metrics, including the Consistency Index, Retention Index, and Rescaled Consistency Index. In these metrics, higher values suggest high-quality feature subsets, whereas lower scores on the Homoplasy Index indicate minimized noise within the feature set. A comprehensive analysis reveals that the algorithm efficiently handles non-linear complexities, enabling it to distinguish between good and bad feature sets while maintaining computational efficiency. The results underscore the importance of informed feature selection in improving computational performance and the accurate reconstruction of evolutionary history. Given its flexibility, our algorithm serves as a robust tool for analyzing the evolution of varied pattern systems, making a significant contribution to the emerging field of Scriptinformatics.

Keywords: Cladistic Analysis; computational selection of features; Pattern systems in script analysis; Phylogenetic studies; Scriptinformatics; Arabic and Aramaic scripts.

1. Introduction

Systematics explores the phylogenetic relationships and evolutionary history of taxa. In other words, it is a polycentric discipline that includes different components of biology and molecular phylogenetics. Systematics involves the study of the complexity in the construction and architecture of taxonomic hierarchies, including identification, classification, and nomenclature. Additionally, it examines the evolutionary relationships that connect all organisms and sheds light on their diverse histories. It also involves studying the distribution of organisms, their ecological preferences, and behavior [1].

The evolutionary modeling of script development can be integrated within the discipline of Scriptinformatics, using computer science technologies. Within this interdisciplinary framework, scripts are considered in aggregates reflective of the hierarchical organization into which living systems have emerged and have been misinterpreted as analogous to levels of biological reality. A taxon in scriptinformatics describes a specific script or the variety of one, with all inscriptions falling into that group. It provides a systematic and progressive way

*Email: osamaalisalman.khafajy@edu.bme.hu

to study the evolution of scripts as well as their spread across cultures [2]. The use of evolutionary analysis in scriptinformatics demonstrates the narrative implications of how human communication has evolved in different historical periods.

In our research, we present a feature selection method aimed at improving the accuracy and efficiency of inference in script systems [3]. This study was initially presented at the IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2024), where we focused on implementing this approach. However, during the analysis, we noted some drawbacks related to processing time and an anticipated challenge in solving a test due to an inherent issue.

In this version, we address these challenges by re-examining the analyses. We also present the time taken across rounds to ensure consistency and precision. Furthermore, we provide a detailed examination of the effective evolutionary tree. Additionally, we include a diagram that illustrates the algorithm's process alongside a graph showing the correlation between the number of selected features and the analysis duration. These elements offer deeper insights into the algorithm's performance in terms of feature selection and time management, aspects that were not covered in the initial conference paper.

Generally, a script in the domain of Scriptinformatics is broadly defined as being encapsulated in a pattern system composed of symbols, syntax, and layout rules specific to symbolic communication methods. These systems are unique in that they possess binary features, noted as either present (1) or absent (0). Pattern evolution studies typically employ a taxon as a pattern system, such as Morse code, microelectronic design layout guidelines, or ancient scripts. Thus, the evolution of writing systems is not just viewed from two perspectives, script-informatics and process-based processing, but from countless levels. These intricate meta-systems are regarded as distinct entities or taxa within pattern systems.

Pattern evolution is the area of research concerned with the evolutionary development of patterns. This field of study is concerned with the evolution and development of different patterns. This diversity can be classified and examined as systems, many of which are separate scripts developed by various human cultures. These scripts evolve and become more complex over time, shaped by a wide range of social and cultural influences. Researchers studying pattern evolution investigate how these recognizable pattern systems or scripts have transformed across different eras and spread through various scripts [4]. These details, in addition to each contributing paper, are important insights revealing how varied writing systems were created and connected with human societies for very long.

Scriptinformatics: The study of symbols, rules, and layouts that have lasted through millennia across scripts, their possible evolution or spread together with the change in logistics, and why some characteristics evolve while others are stagnant over centuries at geographically distant regions/script groups. This area also designs computational algorithms and uses machine learning to deal with large amounts of textual data, enabling the reading of historical scripts like Aramaic, Syriac, and Arabic (Kufic, Naskh, African, and Early) but has allowed people to self-publish their own creation [5]. Scriptinformatics is a novel emerging discipline of the development and impact of writing systems on modern society. A project that combines convolutional neural networks (CNN) with support vector machines for a systematic exploration of the similarities among scripts from various historical epochs [6].

Scriptinformatics is related to research done in [7] and the Internet of digital reality that uses artificial intelligence to improve digital experiences. In the arena of evolutionary modelling, machine learning is used to resolve phylogenetic relationships between taxa. Taxon is a unique characteristic of each item, and the values of this taxa are assigned by another technique named feature engineering. Key features should carefully be selected from relevant properties for effective betterment of phylogenetic analysis.

Feature selection helps identify the most relevant features from vast datasets, which is essential for phylogenetic analysis. Different methods, including filter, wrapper, embedded, and hybrid techniques, are used to overcome related challenges and achieve their advantages [8]. Based on the task and data, the appropriate method is selected to address the given problem statement. When applied effectively these techniques significantly enhance the precision and efficiency of analyses as well as deepen our comprehension of evolutionary relationships among taxa.

The significance of feature selection methods in model development within the field of bioinformatics has been accentuated by the proliferation of high-dimensional data [8,9]. The methodology involves the selection of key features to optimize performance or efficiency, analyzing up to $2^n - 1$ different subsets for n features, which can be condensed to $2^n - 2$ omitting the null set ($\emptyset$). Brute-force techniques can be efficient for handling small datasets with a small amount of features n however, they prove to be impractical due to the substantial size and complexity of any typical datasets in the informatic field.

Numerous feature selection techniques have been created to enhance model performance and efficiency by minimizing data dimensionality and removing irrelevant features. These methods play a crucial role in phylogenetic studies, allowing for creating precise models and offering a more profound understanding of the evolutionary connections among taxa. Despite challenges associated with specific features, many strategies have been devised to optimize phylogenetic reconstruction, enhancing the accuracy and efficiency of analyses.

In this study, we deal with historical scripts, including Arabic, Aramaic, and Middle Iranian scripts, taken as independent pattern systems and classified as taxa. This effort, therefore, tries to find the best algorithm for reconstructing a phylogenetic tree for them, and the dataset is provided with GitHub [10,11,12].

2. Literature Review

2.1 Feature Selection Challenges

Reducing dataset dimensionality, which involves decreasing the number of features, is essential for optimizing and preserving key characteristics, thereby enhancing model accuracy and streamlining the identification of optimal phylogenetic trees. This process not only removes irrelevant or redundant features, increasing computational efficiency, but also highlights significant features that may require further investigation, particularly in phylogenetic reconstructions. Overall, reducing dimensionality speeds up and improves the analysis of complex datasets, deepening our comprehension of evolutionary relationships [13].

Reconstruction of phylogenetic tree using maximum parsimony minimizes genetic changes and branch lengths to produce the most straightforward and plausible trees, highlighting taxa interconnections and simplifying evolutionary narratives [4,14,15].

Feature selection is categorized into filter, wrapper, and embedded methods, each optimizing model construction differently [8,16,9]. These methods enhance model accuracy and efficiency, enabling the precise selection of impactful features, thereby improving analytical outcomes.

Filter methods assess and prioritize features by analyzing their inherent properties, making them ideal for managing high dimensional datasets due to their computational independence and efficiency [8,17]. Despite potential suboptimal performance due to their univariate nature and lack of interaction with classifiers [18], they are essential for initial evaluations and, when combined with other methods, enhance the development of robust classification models.

Wrapper methods combine feature selection with model building by evaluating and selecting features based on their contribution to the model's performance during the training

and testing phases. In this approach, feature selection is designed around the needs of the final model [8,9,16,19]. In contrast, their flexibility leads to broader search spaces and higher risks of overfitting; thus, careful model validation is necessary for accuracy to be ensured [9,16,20].

Embedded techniques simplify feature selection by incorporating it directly into the process of model building. One of these techniques described for certain algorithms allows the model to explore simultaneously all the different feature subsets during training, optimizing the hypothesis space [9,16,21].

Feature selection in phylogenetic models is always based on metrics evaluating feature relevance or importance with respect to inference accuracy and efficiency. The ability of these metrics to identify the most informative features in the construction of optimal phylogenetic trees signifies a role for sophisticated feature selection in enhancing the results of phylogenetic analyses.

2.1 Feature Selection Challenges

To assess the effectiveness of phylogenetic trees derived from phylogenetic inference, indices such as CI, RI, RCI, and HI are utilized, representing the Consistency Index, Retention Index, Rescaled Consistency Index, and Homoplasy Index, respectively.

The CI measures the minimum proportion of feature state changes calculated from dividing the smallest number of required changes by the actual observed changes [15]. This index assesses all aspects of a phylogenetic tree, labeled as τ for each feature j ($where\ j = 1,2,\dots,n$) in the τ tree, an associated state change s_j occurs. Based on this data, a tree is constructed such that the number of changes m_j in the state of the j^{th} feature. The feature is minimized, effectively capturing the simplest evolutionary path possible as in (1).

$$CI(\tau) = \frac{\sum_{j=1}^{n} m_j}{\sum_{j=1}^{n} s_j} \tag{1}$$

The retention index RI quantifies the amount of homoplasy in a given phylogenetic tree τ relative to the maximum possible. The actual number of observed feature state changes is subtracted from the maximum possible state changes, determined by the smaller sum of '1' or '0' states for each feature across all features within the tree. This obtained difference then undergoes division by the range between maximum and minimum feasible state change with respect to the tree, which is elaborated in equation (2) given by [22], [23].

$$RI(\tau) = \sum_{j=1}^{n}\left(M_j - s_j\right)/\sum_{j=1}^{n}\left(M_j - m_j\right) \tag{2}$$

One of the challenges associated with the CI is that its value is influenced by both the number of taxa and the number of features in the dataset, making its interpretation more complex. To simplify this complication, the RCI was developed because the raw values of the CI can easily be normalized on a scale between 0 and 1 by a simple formula. It is therefore easy to interpret, independent of dataset size and complication. It gives the goodness of fit of the phylogenetic tree to the relationships of the data set from which it was constructed. The RC is calculated as the observed CI divided by the minimum possible CI, as specified in equation (3).

$$RC = \frac{CI_{obs}}{CI_{min}} \tag{3}$$

Normalizing CI values allows for comparative assessments of tree fit across different studies or models. Similarly, the HI is applied in cladistic analysis to quantify the degree of

homoplasy within an evolutionary tree. It is calculated as the ratio of observed homoplasious features to the maximum possible in the dataset, as outlined in equation (4).

$$HI = \frac{Number\ of\ Homoplasious\ Features}{Max.Possible\ Homoplasious\ Features} \times 100 \quad (4)$$

2.3 Application of Maximum Parsimony and Feature Selection

In our Maximum Parsimony analysis, the algorithm used was Branch and Bound, which cuts branches optimally without the best solution to find the most suitable tree [1]. The reason for using the Branch and Bound method is that our dataset includes 19 taxa and an exhaustive search for the most parsimonious tree is not computationally feasible without optimization. This approach will certainly return the Maximum Parsimony tree without unnecessary processing and exploration of inferior branches, thereby drastically improving computation efficiency while guaranteeing high-quality results.

3 Methodology

The proposed approach seeks to enhance the efficiency of phylogenetic tree reconstruction by positing that similar features are likely to converge toward a global maximum in Maximum Parsimony analysis. To validate this hypothesis, the dataset, which consists of various taxa, undergoes preprocessing where duplicate features are removed based on their similarity. This results in a more streamlined dataset that retains only unique taxonomic features, as described in Algorithm 1. By refining the dataset in this way, we enhance both the accuracy and computational efficiency of the subsequent analyses.

Algorithm 1: TSSF (Taxonomic Similarity-based feature Selection and Filtration)

Step 1: Input Declaration
Inputs:

- **Xα**: Original dataset containing taxonomic features.
- **Numerical_F**: A matrix representation of extracted numerical features.
- **T_Name**: A cell array storing taxonomic labels.
- **Ldata**: A dataset representing pairwise Euclidean distances between features.
- Thresholds (**upThr**): A predefined set of similarity thresholds.
- **MuB**: Maximum distance found in **uB**.

Step 2: Data Preprocessing
Purpose: Compute distances and remove duplicate features to reduce dataset complexity.
Processing Steps:

1. Compute **pairwise Euclidean distances** among features in Xα.
2. Store the calculated distances in **uB**.
3. Identify and remove identical features **(distance = 0)** while keeping only one representative.
4. Save the **reduced dataset** as Xβ (cleaned and unique feature set).

Step 3: Initialize Feature Selection Parameters
Purpose: Prepare for iterative feature selection.
Processing Steps:

1. Set the **initial similarity threshold**:
    - upThr = 1
    - MuB = max(uB) (Maximum distance in uB).
2. **Begin iterative feature filtration.**

Step 4: Iterative Feature Selection (Loop Processing)
Purpose: Incrementally filter redundant features by iterating through similarity thresholds.
Processing Steps:

1. **Repeat while upThr ≤ MuB:**
    - Compute **pairwise Euclidean distances** among features in Xβ.
    - Store calculated distances in uB.
    - Identify **features with distances ≤ upThr**.
    - **Remove redundant features**, keeping only one per similarity threshold.
    - Store the **filtered dataset** at this threshold in tD.
    - **Increment upThr (upThr = upThr + 1)** and go back to step **4.1** to re-evaluate.
2. **If upThr > MuB, exit the loop and proceed to output.**

Step 5: Output Declaration
Outputs:

- **Final refined dataset (tD)** containing only unique and non-redundant features.
- **Archived filtered datasets** at different similarity thresholds for comparative analysis.
- **Final dataset (Xβ)**, optimized for phylogenetic processing.

Step 6: Phylogenetic Analysis
The **refined dataset (tD)** is used for **full genetic analysis**, ensuring **optimized feature selection** for phylogeny reconstruction.

Euclidean distances are used to evaluate the relationships among the features, identifying and eliminating matched or nearly matched ones. The overall methodology behind our feature selection algorithm is visually represented in Figure 1. The algorithm starts with the important step of the Preprocessing block, where the raw input undergoes systematic refinement in a preparatory stage that means data processing, cleaning the raw data for duplicate features, and generating a streamlined dataset ready for in-depth phylogenetic examination.

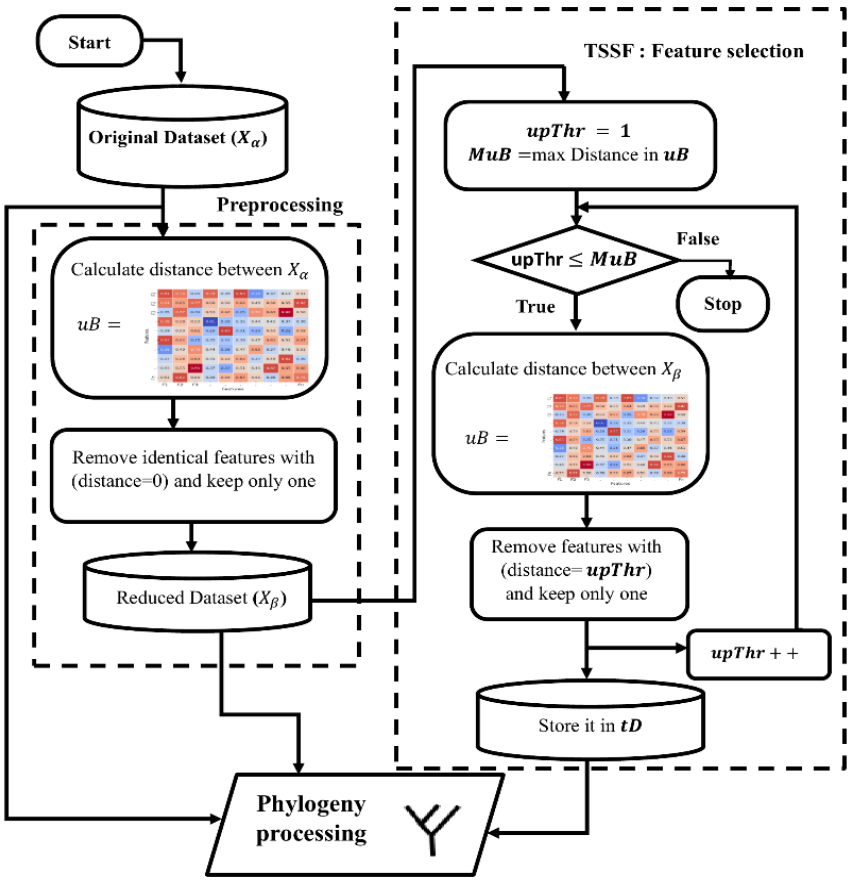

Figure 1: TSSF: Taxonomic Similarity-based Feature Selection and Filtration Algorithm Flowchart

After the preprocessing step, the centerpiece of the algorithm is the Feature Selection block. In this step, iterative filtration of features is done by using strict threshold values on the relevance of each feature. This step has great importance since it selects only those features that contribute significantly to the task of phylogenetic tree reconstruction. The process aims at not only maximizing the accuracy of the analysis but also ensuring computational efficiency. This approach is important because the interplay between the pre-processing and feature selection stages encodes the robustness of the TSSF algorithm. The effectiveness of preprocessing directly brings about success in feature selection, since more accurate and computationally efficient results in the phylogenetic analysis result after feature selection.

4 Results

We conducted a performance assessment of our method for identifying the global optimum in phylogenetic trees using 19 taxa with varying attributes. This enabled the efficient application of Maximum Parsimony analysis. The results of various tests involving unique

feature selection approaches, including the elimination of features based on Euclidean distance, are summarized in Table 2.

Table 2 presents key performance indicators, including the count of maximum parsimony trees identified, the number of features chosen tree length, and several critical phylogenetic metrics, including the CI, RI, RC, and HI. The first row ($upThr = non$) serves as a control case where no feature selection was applied, providing a baseline for comparison. The 'Average Time' column was added to assess the computational effort required, with each entry representing the mean of five simulation runs per test.

Table 2: Impacts of Diverse Feature Selection Approaches in Maximum Parsimony Analysis

Test.	$upThr$	Optimal Trees found	Feature No.	Length	CI	RI	rescaled CI	HI	Average Time
1	non	2	97	229	0.424	0.615	0.261	0.576	0:05:56.00
2	0	4	73	197	0.371	0.585	0.217	0.629	0:10:21.46
3	1	5	53	160	0.331	0.577	0.191	0.669	0:22:34.84
4	2	12	49	145	0.338	0.57	0.192	0.662	0:29:58.06
5	3	3	32	97	0.33	0.586	0.193	0.67	0:10:28.34
6	4	6	15	36	0.417	0.7	0.292	0.583	0:00:02.78
7	5	7	14	33	0.424	0.708	0.3	0.576	0:00:03.35
8	6	1	7	13	0.538	0.824	0.443	0.462	0:00:22.26
9	7	6	6	7	0.857	0.988	0.81	0.143	0:03:17.22
10	8	48	10	19	0.526	0.75	0.395	0.474	0:00:00.60
11	9	6	6	9	0.667	0.86	0.576	0.333	5:22:41.04
12	10	1906	13	21	0.619	0.733	0.454	0.381	0:00:24.10
13	11	32	19	38	0.5	0.678	0.339	0.5	0:00:00.52
14	12	209	18	38	0.474	0.661	0.313	0.526	0:00:07.00
15	13	226	28	67	0.418	0.571	0.239	0.582	0:03:34.06
16	14	2	39	95	0.411	0.573	0.235	0.589	0:01:42.26
17	15	14	50	132	0.379	0.539	0.204	0.621	0:19:03.34
18	16	6	58	157	0.369	0.554	0.205	0.631	0:29:12.68
19	17	10	64	176	0.364	0.573	0.208	0.636	0:23:41.94
20	18	6	69	187	0.369	0.579	0.213	0.631	0:12:34.72

This study conducted a chain of tests to evaluate the effectiveness of the TSSF method in generating accurate cladograms. Test 1, with no feature deletions, produced 229 Trees Length with a moderate $CI = 0.424$. Subsequent tests applied varying thresholds for feature deletion, influencing the number and structure of resulting trees. Notably, higher thresholds ($upThr$) typically yielded fewer trees but more optimal solutions. For example, Test 7 resulted in a high $RI = 0.708$ and a low $RC = 0.300$ demonstrating a better fit. Test 10 also showed a moderate $CI = 0.526$ and an $RC = 0.395$.

Reducing the number of features too much, especially in tests 7–12, led to cladograms with unresolved relationships. However, the consistency in average clade evaluation consistency across tests indicates that the TSSF method effectively identified key features that maintained important evolutionary relationships. The computational time required for Test 11 was substantial, taking approximately 5 and a half hours, as shown in Table 2. The study highlighted the importance of maintaining a careful balance between feature selection and tree topology to ensure stable subsets.

For visual representations of these findings [12], a set of cladograms illustrating the phylogenetic relationships among historical scripts and their variants is provided. Figure 2 displays the cladograms for Test 1 and Test 19, labeled as (a) and (b), respectively, demonstrating the effectiveness of the developed approach.

The computation time for this test 11 in Table 2 was significant, amounting to approximately 5 and a half hours. This outcome has been integrated into the broader analysis, highlighting the substantial computational effort required for certain feature selection scenarios. The updated data now more accurately reflects the computational needs of the algorithm, as illustrated in Figure 3.

The experimental findings emphasize the significant role of feature selection in Maximum Parsimony analysis. Furthermore, these results suggest that the proposed method is capable of identifying optimal solutions that are specifically tailored to the characteristics of different datasets. The findings offer valuable insights into the impact of feature similarity on the accuracy of the Maximum Parsimony method.

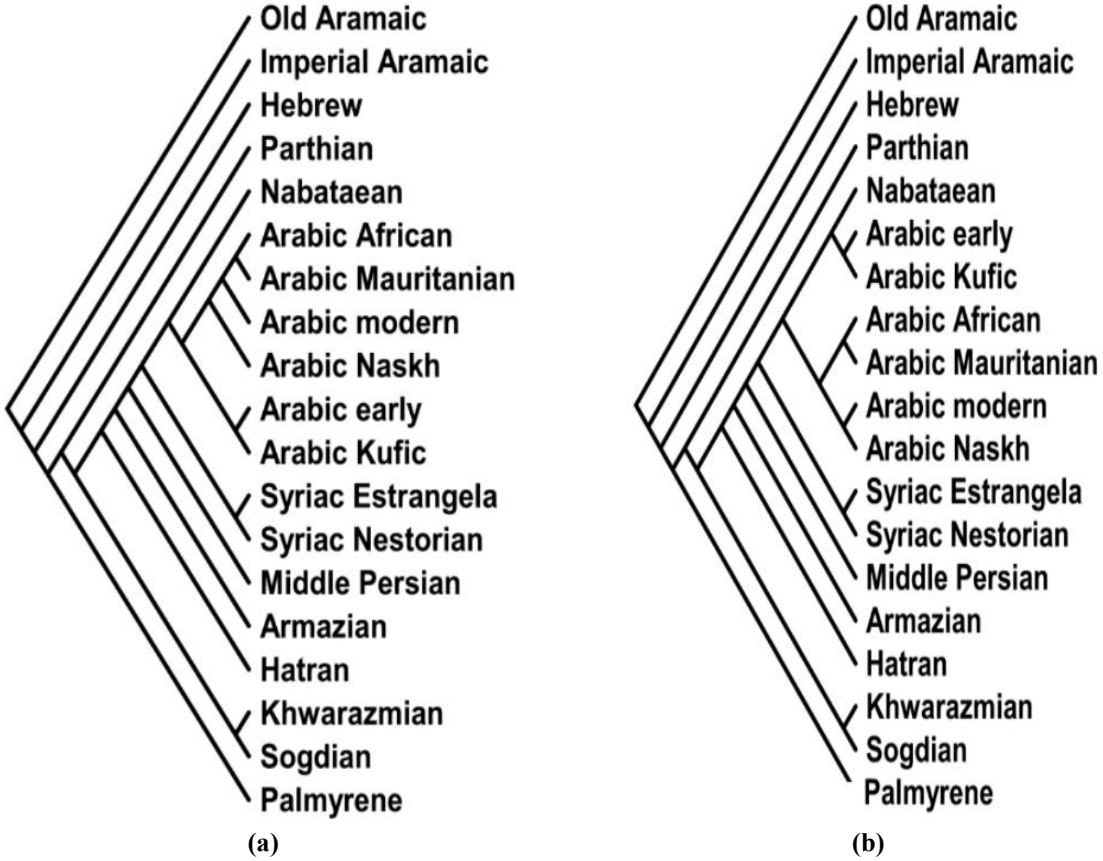

Figure 2: The chosen phylogenetic cladograms showcase the historical scripts and their variants, with Test 1 being illustrated by (a) and Test 19 by (b)

By utilizing the PAUP software, the suggested method generated results in the form of phylogenetic trees (cladograms) and correlated measurements in our case Tree Length, Consistency Index (CI), Retention Index (RI), Rescaled Consistency Index (RC), and Homoplasy Index (HI).

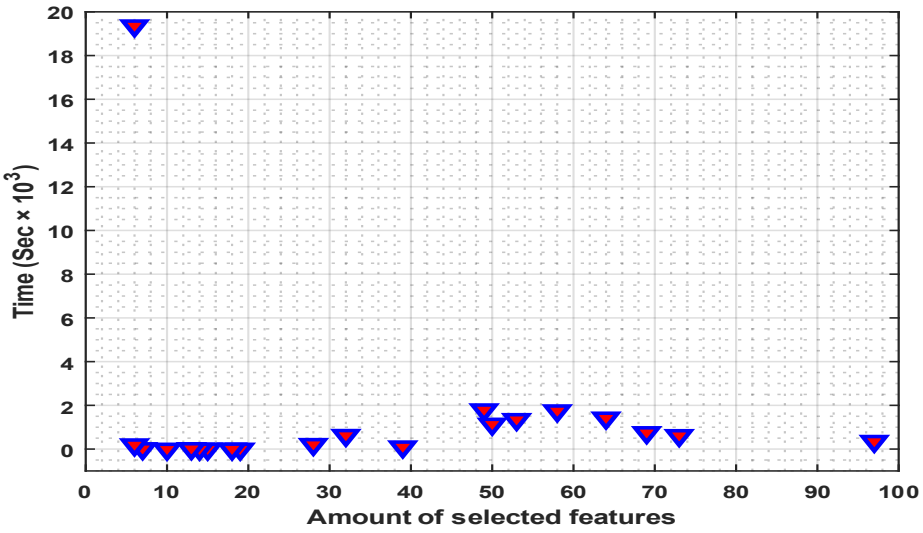

Figure -3 TSSF Performance: Time vs. Number of Selected Features for Maximum Parsimony Tree Construction

Besides feature subset selection, a very important factor that had to be taken into account in our phylogenetic study was the computation time that each test required. Figure 3 shows the average computation time with interesting relationships for the number of features.

In contrast to our initial assumption, we found that lowering the number of features does not necessarily mean shorter computation time. Specific tests demonstrate this non-linear relationship. Here, for example, you could observe that while Tests 9 and 11 both contained same number of features, they still had a large difference between their average computation times. This is due to the difference between *upThr* in Table 2, which shows how seriously the choice of FS criteria can influence the computational cost.

Additionally, where Test 10 is concerned, we can see that the calculation time drops drastically within a normal number of features. This is most likely due to the behavior of the Branch-and-Bound algorithm we employ in our work. This algorithm could easily search for Maximum Parsimony trees, but it would sometimes become effectively exhaustive as far as branches are concerned.

These observations show that the dichotomy in phylogenetic analysis is only part of what makes real data complex. The interaction between the number of features and search algorithm dynamics is critical. This not only affects the computational time but also strengthens and affects the more general applicability of it. Consequently, our study illustrates the necessity of integrating feature selection and computational efficiency to conduct complete phylogenetic research.

5 Conclusions

This study assesses the efficacy of the TSSF (Taxonomic Similarity-based feature Selection and Filtration) method in optimizing phylogenetic tree reconstruction using Maximum Parsimony analysis through the application of diverse preprocessing techniques and a feature selection process based on Euclidean distance.

The initial assumption regarding the outcome of Test 11 was that it had encountered an underflow, given the considerable time required for its completion. Further investigation showed that the Branch-and-Bound algorithm, used to perform the search for an optimal parsimonious tree, was only operating linearly because of the inordinate number of branches being required to explore by the features chosen. In this test, it was shown just how intense some feature selection scenarios must be computationally. It also brought out the point that

feature selection plays an important role in managing computational resources because an exhaustive search would be impossible in such situations.

The study further examined the relationship between the number of features and the computation time required, identifying complex patterns that challenge traditional assumptions about computational efficiency in phylogenetic analysis. The TSSF method, which belongs to the filter category of feature selection, evaluates features based on their inherent properties, specifically their pairwise Euclidean distances. Our results indicate that a trade-off in feature optimization must be carefully managed between rich tree structure representation and computational feasibility. Striking the right balance is crucial, as demonstrated in the tests where feature reduction improved computational time and maintained the integrity of the phylogenetic trees.

Acknowledgment

Gratefully acknowledged is the support of Stipendium Hungaricum Scholarship which in no small way gave invaluable assistance and directly contributed to the implementation of our research work. This opportunity with such a highly esteemed scholarship program played a vital role in our academic development and in bringing the work to a successful completion.

6. Disclosure and conflict of interest

Conflict of Interest: The authors declare that they have no conflicts of interest.

References

[1] I. J. Kitching, P. Forey, C. Humphries, and D. Williams, *Cladistics: The Theory and Practice of Parsimony Analysis*. Oxford, U.K.: Oxford Univ. Press, 1998.

[2] J. D. Washburn, K. A. Bird, G. C. Conant, and J. C. Pires, "Convergent evolution and the origin of complex phenotypes in the age of systems biology," *Int. J. Plant Sci.*, vol. 177, no. 4, pp. 305–318, 2016, doi: 10.1086/686009.

[3] O. A. Salman and G. Hosszú, "Phylogenetic inference using advanced feature selection," in *Proc. 14th IEEE Int. Conf. Cognitive Infocommunications (CogInfoCom)*, 2023, pp. 000173–000178, doi: 10.1109/CogInfoCom2023.10397530.

[4] O. A. Salman and G. Hosszú, "Cladistic analysis of the evolution of some Aramaic and Arabic script varieties," *Int. J. Appl. Evol. Comput.*, vol. 12, no. 4, pp. 18–38, 2021, doi: 10.4018/IJAEC.2021100103.

[5] O. A. Salman, G. Hosszú, and F. Kovács, "A new feature selection algorithm for phylogenetic analysis of Aramaic and Arabic script variants," *Int. J. Intell. Eng. Inform.*, vol. 10, no. 4, pp. 313–331, 2022, doi: 10.1504/IJIEI.2022.128892.

[6] S. Daggumati and P. Z. Revesz, "Convolutional Neural Networks analysis reveals three possible sources of Bronze Age writings between Greece and India," *Information*, vol. 14, art. no. 227, 2023, doi: 10.3390/info14040227.

[7] P. Baranyi, A. Csapo, and G. Sallai, *Cognitive Infocommunications (CogInfoCom)*. Springer, 2015.

[8] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007, doi: 10.1093/bioinformatics/btm344.

[9] N. Ansari, "A survey on feature selection techniques using evolutionary algorithms," *Iraqi Journal of Science*, vol. 62, no. 8, pp. 2796–2812, 2021, doi: 10.24996/ijs.2021.62.8.32.

[10] O. A. Salman and G. Hosszú, *Arabic-Aramaic-DataSet*, GitHub, 2021–2022 [cited 2024 Sep 21]. Available: https://github.com/OsamaAliSalman/Arabic-Aramaic-DataSet.

[11] O. A. Salman and G. Hosszú, *Extended_Arabic-Aramaic-DataSet*, GitHub, 2024 [cited 2024 Sep 21]. Available: https://github.com/OsamaAliSalman/Extended_Arabic-Aramaic-DataSet.

[12] O. A. Salman, *script-evolution-cladograms*, GitHub, [cited 2024 Sep 21]. Available:

https://github.com/OsamaAliSalman/script-evolution-cladograms.

[13] O. A. Salman and G. Hosszú, "Optimised feature dimension reduction method and its impact on the search for optimal trees," in *Proc. Workshop on the Advances of Inf. Technol. (WAIT 2023)*, B. Kiss and L. Szirmay-Kalos, Eds., Budapest: BME, 2023, pp. 23–28.

[14] L. Kannan and W. C. Wheeler, "Maximum parsimony on phylogenetic networks," *Algorithms Mol*. Biol., vol. 7, no. 1, pp. 1–10, 2012, doi: 10.1186/1748-7188-7-9.

[15] A. G. Kluge and J. S. Farris, "Quantitative phyletics and the evolution of the Anurans," *Syst. Zool.*, vol. 18, no. 1, pp. 1–32, 1969, doi: 10.2307/2412407.

[16] A. S. Issa, Y. H. Ali, and T. A. Rashid, "Review on hybrid swarm algorithms for feature selection," *Iraqi Journal of Science*, vol. 64, no. 10, pp. 5331–5344, 2023, doi: 10.24996/ijs.2023.64.10.38.

[17] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowé, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 9, no. 4, pp. 1106–1119, 2012, doi: 10.1109/TCBB.2012.33.

[18] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, 2004.

[19] M. Shardlow, "An analysis of feature selection techniques," *The University of Manchester*, vol. 1, pp. 1–7, 2016.

[20] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra, "Feature subset selection by Bayesian network-based optimization," *Artif. Intell.*, vol. 123, no. 1–2, pp. 157–184, 2000, doi: 10.1016/S0004-3702(00)00052-7.

[21] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *J. Mach. Learn. Res.*, vol. 3, pp. 1439–1461, 2003, doi: 10.1162/153244303322753751.

[22] F. J. Farris, "The retention index and the rescaled consistency index," *Cladistics*, vol. 5, no. 4, pp. 417–419, 1989, doi: 10.1111/j.1096-0031.1989.tb00573.x.

[23] D. Lipscomb, Basics of Cladistic Analysis, Washington, D.C.: *George Washington University*, 1998 [cited 2024 Sep 21]. Available: https://www2.gwu.edu/~clade/faculty/lipscomb/Cladistics.pdf